
Python YouTrack Documentation

Release 0.7.1+1.g3af4648.dirty

Ali-Akber Saifee

Oct 29, 2017

Contents

1	Contents	3
2	Command line usage	23
	Python Module Index	35

Command line utility to interface with YouTrack

- Free software: MIT license
- Documentation: <https://pyutrack.readthedocs.io>.

1.1 Command line usage

The cli is invoked using the `pyu` executable.

1.1.1 Configuration

Before getting started you should configure `pyu`. This can be done manually by creating an ini file at `~/ .pyutrack` with the following contents:

```
[pyutrack]
base_url = https://my.myjetbrains.com/youtrack/
username = myusername
password = mypassword
```

`password` is optional and if not provided will be prompted for and subsequently persisted to the system keychain.

You can optionally store the config file in a custom location and on subsequent invocations of the executable provide the `--config=${CONFIG_LOCATION}` option.

Lastly, you can interactively generate the configuration by calling `pyu new config`

1.1.2 Quick start

All interaction with YouTrack resources is performed around five main sub-commands: `new`, `update`, `list`, `show` & `delete`.

Here's a few examples to get started. For more details on each sub-command refer to *command documentation*.

Listing...

Users

```
pyu list users
```

Projects

```
pyu list projects
```

All Issues in a project

```
pyu list issues --project=MYPROJECT
```

Issues filtered with a query (Search Query Reference)

```
pyu list issues --filter='crash' # issues containing the word "crash"  
pyu list issues --filter='for: me' # issues for current user  
pyu list issues --filter='reporter: me' # issues reported by current user  
pyu list issues --filter='priority: Critical' # critical issues
```

Creating...

New regular user

```
pyu new user new_user01 'New User' newuser@moo.com password
```

New user with specific group(s)

```
pyu new user new_admin01 'New Admin' newadmin@moo.com password --group=Admin
```

New issue

```
pyu new issue MYPROJECT 'this is an important issue'
```

New issue tagged with kitties and marked as critical and assigned to me (Command Reference)

```
pyu new issue MYPROJECT 'this is an important issue' --command='tag kitties priority_↵  
↵critical assignee me'
```


Updating...

Change a user's password

```
pyu update user some-user --password=new-password
```

Add a comment to an issue:

```
pyu update issue PRO-1 --comment='test comment'
```

Update an issue using a command (Command Reference)

```
pyu update issue PRO-1 --command='assignee: me' # assign issue to yourself
pyu update issue PRO-1 --command='tag: kitties' # tag the issue with kitties
pyu update issue PRO-1 --command='priority: critical' # increase priority to critical
pyu update issue PRO-1 --command='fixed' # mark issue as fixed
```

1.1.3 Command documentation

pyu

YouTrack command line interface

```
pyu [OPTIONS] COMMAND [ARGS]...
```

Options

- base_url** <base_url>
root url of your youtrack installation
- username** <username>
username to access youtrack as
- password** <password>
password for current user
- debug, --no-debug**
enable/disable verbose logging
- watch** <watch>
watch the output of the command and run repeatedly every *n* seconds

delete

delete youtrack resources

```
pyu delete [OPTIONS] COMMAND [ARGS]...
```

group

```
pyu delete group [OPTIONS] NAME
```

Arguments

NAME

Required argument

issue

```
pyu delete issue [OPTIONS] ID
```

Arguments

ID

Required argument

project

```
pyu delete project [OPTIONS] ID
```

Arguments

ID

Required argument

user

```
pyu delete user [OPTIONS] LOGIN
```

Arguments

LOGIN

Required argument

list

List various youtrack resources

```
pyu list [OPTIONS] COMMAND [ARGS]...
```

groups

list groups [Admin only]

```
pyu list groups [OPTIONS]
```

Options

--user <user>
user login

issues

list issues

```
pyu list issues [OPTIONS]
```

Options

--project <project>
--filter <filter>
--limit <limit>

links

list issue links

```
pyu list links [OPTIONS]
```

Options

--issue <issue>
issue id

permissions

list permissions [Admin only]

```
pyu list permissions [OPTIONS]
```

Options

--role <role>
role name

projects

list projects [Admin only]

```
pyu list projects [OPTIONS]
```

roles

list roles [Admin only]

```
pyu list roles [OPTIONS]
```

Options

--group <group>
group name

--user <user>
user login

users

list users [Admin only]

```
pyu list users [OPTIONS]
```

Options

--group <group>
group name

--role <role>
role name

--query <query>
query to match users against

--permission <permission>
permission name

--project <project>

new

create new youtrack resources

```
pyu new [OPTIONS] COMMAND [ARGS]...
```

config

generate a new config

```
pyu new config [OPTIONS]
```

Options

--force <force>

--path <path>

group

create a new group [Admin only]

```
pyu new group [OPTIONS] NAME
```

Options

--description <description>

--auto-join, --no-auto-join
automatically add new users

Arguments

NAME

Required argument

issue

create a new issue

```
pyu new issue [OPTIONS] PROJECT SUMMARY
```

Options

--description <description>
issue details

--command <command>
command to run after issue is created

Arguments

PROJECT

Required argument

SUMMARY

Required argument

project

create a new project [Admin only]

```
pyu new project [OPTIONS] ID NAME
```

Options

--lead <lead>

Arguments

ID

Required argument

NAME

Required argument

role

create a new role [Admin only]

```
pyu new role [OPTIONS] NAME
```

Options

--description <description>

--permission <permission>
permission(s) to give new role

Arguments

NAME

Required argument

user

create a new user [Admin only]

```
pyu new user [OPTIONS] LOGIN NAME EMAIL PASSWORD
```

Options

--group <group>
group(s) to add new user to

Arguments

LOGIN
Required argument

NAME
Required argument

EMAIL
Required argument

PASSWORD
Required argument

show

show details for youtrack resources

```
pyu show [OPTIONS] COMMAND [ARGS]...
```

config

show current cli configuration

```
pyu show config [OPTIONS]
```

group

show group details [Admin only]

```
pyu show group [OPTIONS] NAME
```

Arguments

NAME
Required argument

issue

show issue details

```
pyu show issue [OPTIONS] ID
```

Arguments

ID

Required argument

project

show project details [Admin only]

```
pyu show project [OPTIONS] ID
```

Arguments

ID

Required argument

role

show role details [Admin only]

```
pyu show role [OPTIONS] NAME
```

Arguments

NAME

Required argument

user

show user details [Admin only]

```
pyu show user [OPTIONS] LOGIN
```

Arguments

LOGIN

Required argument

update

update existing youtrack resources

```
pyu update [OPTIONS] COMMAND [ARGS]...
```

group

update a group [Admin only]

```
pyu update group [OPTIONS] NAME
```

Arguments

NAME

Required argument

issue

update an issue

```
pyu update issue [OPTIONS] ID
```

Options

--summary <summary>

--description <description>

--command <command>
command to apply to issue.

--comment <comment>
comment to add to issue

Arguments

ID

Required argument

role

update a role [Admin only]

```
pyu update role [OPTIONS] NAME
```

Options

+permission <add_permissions>
add permission

-permission <remove_permissions>
remove permission

Arguments

NAME
Required argument

user

update a user [Admin only]

```
pyu update user [OPTIONS] LOGIN
```

Options

--name <name>

--email <email>

--password <password>

+group <add_groups>
add user to group

-group <remove_groups>
remove user from group

Arguments

LOGIN
Required argument

1.2 Installation

1.2.1 Stable release

To install YouTrack CLI, run this command in your terminal:

```
$ pip install pyutrack
```

This is the preferred method to install YouTrack CLI, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

1.2.2 From sources

The sources for YouTrack CLI can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/alisaiffee/pyutrack
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/alisaiffee/pyutrack/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

1.3 API

The internal API is **internal** at the moment and - though it can be used for interacting with the YouTrack REST API using an object oriented interface - is not supported.

1.3.1 Connection

```
class pyutrack.connection.Connection (base_url=None, credentials=None)
```

```
    api_url  
    base_url  
    credentials  
    delete (*args, **kwargs)  
    get (*args, **kwargs)  
    login (persist_credentials=True)  
    post (*args, **kwargs)  
    put (*args, **kwargs)
```

```
class pyutrack.connection.Credentials (username, password=None, cookies=None)
```

```
    KEYRING_COOKIE = 'pyutrack:cookies'  
    KEYRING_PASSWORD = 'pyutrack:password'  
    cookies  
    static from_keyring (username)  
    load_from_keyring ()  
    password  
    persist ()  
    reset_cookies ()
```

```
reset_password()
```

```
username
```

```
pyutrack.connection.fix_auth(meth)
```

1.3.2 Resource Types

```
class pyutrack.resources.Group(connection, hydrate=False, **fields)
```

```
classmethod create(connection, name, auto_join=False, description="")
    create group
```

```
delete()
    delete group
```

```
get()
    get group
```

```
get_roles()
    get group roles
```

```
classmethod list(connection)
    list group
```

```
roles
    group roles
```

```
update(description, auto_join, new_name)
    update group
```

```
class pyutrack.resources.Issue(connection, hydrate=False, **fields)
```

```
assignee
```

```
command(command=None, comment=None)
    executes a command for the given issue.
```

Parameters `command` (*str*) – The youtrack command to execute. See <https://www.jetbrains.com/help/youtrack/standalone/Commands.html> for command grammar.

```
classmethod create(connection, project, description="", summary="")
    create issue
```

```
delete()
    delete issue
```

```
get()
    get issue
```

```
get_issue_links()
    get issue issue_links
```

```
id
```

```
issue_links
    issue issue_links
```

```
link
```

```
classmethod list(connection, filter, max=100)
    list issue
```

priority

reporter

status

update (*description=""*, *summary=""*)

update issue

updater

class pyutrack.resources.**IssueLink** (*connection*, *hydrate=False*, ***fields*)

class pyutrack.resources.**IssueLinkType** (*connection*, *hydrate=False*, ***fields*)

classmethod create (*connection*, *outward_name*, *inward_name*, *directed=False*)

create issuelinktype

delete ()

delete issuelinktype

get ()

get issuelinktype

classmethod list (*connection*)

list issuelinktype

update (*new_name=""*, *inward_name=""*, *directed=""*, *outward_name=""*)

update issuelinktype

class pyutrack.resources.**Permission** (*connection*, *hydrate=False*, ***fields*)

classmethod list (*connection*)

list permission

class pyutrack.resources.**Project** (*connection*, *hydrate=False*, ***fields*)

classmethod create (*connection*, *project_name*, *project_lead_login*, *project_id=None*, *description=""*, *starting_number=1*)

create project

delete ()

delete project

description

get ()

get project

get_issues (*filter=""*, *max=100*)

get project issues

id

issues

project issues

lead

classmethod list (*connection*)

list project

```
class pyutrack.resources.Role (connection, hydrate=False, **fields)

    classmethod create (connection, name, description="")
        create role

    delete ()
        delete role

    get ()
        get role

    get_permissions ()
        get role permissions

    classmethod list (connection)
        list role

    permissions
        role permissions

    update (description, new_name)
        update role

class pyutrack.resources.User (connection, hydrate=False, **fields)

    classmethod create (connection, login, full_name, email, password)
        create user

    delete ()
        delete user

    get ()
        get user

    get_groups ()
        get user groups

    get_roles ()
        get user roles

    groups
        user groups

    classmethod list (connection, project="", query="", role="", group="", permission="")
        list user

    roles
        user roles

    update (login="", password="", full_name="", email="")
        update user
```

1.4 Contributing

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

You can contribute in many ways:

1.4.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/alisaiffee/pyutrack/issues>.

If you are 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.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/alisaiffee/pyutrack/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 contributions are welcome :)

1.4.2 Get Started!

Ready to contribute? Here’s how to set up *pyutrack* for local development.

1. Fork the *pyutrack* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/pyutrack.git
```

3. Install your local copy into a virtualenv. Assuming you have *virtualenvwrapper* installed, this is how you set up your fork for local development:

```
$ mkvirtualenv pyutrack
$ cd pyutrack/
$ pip install -r requirements/test.txt
```

4. Create a branch for local development:

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

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests:

```
$ flake8 youtrack_cli tests
$ nosetests
```

6. 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
```

7. Submit a pull request through the GitHub website.

1.4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/alisaiffee/pyutrack/pull_requests and make sure that the tests pass for all supported Python versions.

1.4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_youtrack_cli
```

1.5 Credits

1.5.1 Development Lead

- Ali-Akber Saifee <ali@indydevs.org>

1.5.2 Contributors

None yet. Why not be the first?

1.6 Changelog

1.6.1 0.7.1 2017-10-29

- Fix stack overflow issue with watch

1.6.2 0.7.0 2017-09-21

- Fix create resource

1.6.3 0.6.0 2017-09-14

- Support adding issue comments
- Bug fixes

1.6.4 0.5.0 2017-09-13

- Add watch option

1.6.5 0.4.0 2017-09-06

- First release on PyPI.

Command line usage

The cli is invoked using the `pyu` executable.

2.1 Configuration

Before getting started you should configure `pyu`. This can be done manually by creating an ini file at `~/ .pyutrack` with the following contents:

```
[pyutrack]
base_url = https://my.myjetbrains.com/youtrack/
username = myusername
password = mypassword
```

`password` is optional and if not provided will be prompted for and subsequently persisted to the system keychain.

You can optionally store the config file in a custom location and on subsequent invocations of the executable provide the `--config=${CONFIG_LOCATION}` option.

Lastly, you can interactively generate the configuration by calling `pyu new config`

2.2 Quick start

All interaction with YouTrack resources is performed around five main sub-commands: `new`, `update`, `list`, `show` & `delete`.

Here's a few examples to get started. For more details on each sub-command refer to [command documentation](#).

2.2.1 Listing...

Users

```
pyu list users
```

Projects

```
pyu list projects
```

All Issues in a project

```
pyu list issues --project=MYPROJECT
```

Issues filtered with a query (Search Query Reference)

```
pyu list issues --filter='crash' # issues containing the word "crash"  
pyu list issues --filter='for: me' # issues for current user  
pyu list issues --filter='reporter: me' # issues reported by current user  
pyu list issues --filter='priority: Critical' # critical issues
```

2.2.2 Creating...

New regular user

```
pyu new user new_user01 'New User' newuser@moo.com password
```

New user with specific group(s)

```
pyu new user new_admin01 'New Admin' newadmin@moo.com password --group=Admin
```

New issue

```
pyu new issue MYPROJECT 'this is an important issue'
```

New issue tagged with kitties and marked as critical and assigned to me (Command Reference)

```
pyu new issue MYPROJECT 'this is an important issue' --command='tag kitties priority_  
->critical assignee me'
```

2.2.3 Updating...

Change a user's password

```
pyu update user some-user --password=new-password
```

Add a comment to an issue:

```
pyu update issue PRO-1 --comment='test comment'
```

Update an issue using a command (Command Reference)

```
pyu update issue PRO-1 --command='assignee: me' # assign issue to yourself
pyu update issue PRO-1 --command='tag: kitties' # tag the issue with kitties
pyu update issue PRO-1 --command='priority: critical' # increase priority to critical
pyu update issue PRO-1 --command='fixed' # mark issue as fixed
```

2.3 Command documentation

2.3.1 pyu

YouTrack command line interface

```
pyu [OPTIONS] COMMAND [ARGS]...
```

Options

- base_url** <base_url>
root url of your youtrack installation
- username** <username>
username to access youtrack as
- password** <password>
password for current user
- debug, --no-debug**
enable/disable verbose logging
- watch** <watch>
watch the output of the command and run repeatedly every *n* seconds

delete

delete youtrack resources

```
pyu delete [OPTIONS] COMMAND [ARGS]...
```

group

```
pyu delete group [OPTIONS] NAME
```

Arguments

NAME

Required argument

issue

```
pyu delete issue [OPTIONS] ID
```

Arguments

ID

Required argument

project

```
pyu delete project [OPTIONS] ID
```

Arguments

ID

Required argument

user

```
pyu delete user [OPTIONS] LOGIN
```

Arguments

LOGIN

Required argument

list

List various youtrack resources

```
pyu list [OPTIONS] COMMAND [ARGS]...
```

groups

list groups [Admin only]

```
pyu list groups [OPTIONS]
```

Options

--user <user>
user login

issues

list issues

```
pyu list issues [OPTIONS]
```

Options

--project <project>
--filter <filter>
--limit <limit>

links

list issue links

```
pyu list links [OPTIONS]
```

Options

--issue <issue>
issue id

permissions

list permissions [Admin only]

```
pyu list permissions [OPTIONS]
```

Options

--role <role>
role name

projects

list projects [Admin only]

```
pyu list projects [OPTIONS]
```

roles

list roles [Admin only]

```
pyu list roles [OPTIONS]
```

Options

--group <group>
group name

--user <user>
user login

users

list users [Admin only]

```
pyu list users [OPTIONS]
```

Options

--group <group>
group name

--role <role>
role name

--query <query>
query to match users against

--permission <permission>
permission name

--project <project>

new

create new youtrack resources

```
pyu new [OPTIONS] COMMAND [ARGS]...
```


config

generate a new config

```
pyu new config [OPTIONS]
```

Options

--force <force>

--path <path>

group

create a new group [Admin only]

```
pyu new group [OPTIONS] NAME
```

Options

--description <description>

--auto-join, --no-auto-join
automatically add new users

Arguments

NAME

Required argument

issue

create a new issue

```
pyu new issue [OPTIONS] PROJECT SUMMARY
```

Options

--description <description>
issue details

--command <command>
command to run after issue is created

Arguments

PROJECT

Required argument

SUMMARY

Required argument

project

create a new project [Admin only]

```
pyu new project [OPTIONS] ID NAME
```

Options

--lead <lead>

Arguments

ID

Required argument

NAME

Required argument

role

create a new role [Admin only]

```
pyu new role [OPTIONS] NAME
```

Options

--description <description>

--permission <permission>
permission(s) to give new role

Arguments

NAME

Required argument

user

create a new user [Admin only]

```
pyu new user [OPTIONS] LOGIN NAME EMAIL PASSWORD
```

Options

--group <group>
group(s) to add new user to

Arguments

LOGIN
Required argument

NAME
Required argument

EMAIL
Required argument

PASSWORD
Required argument

show

show details for youtrack resources

```
pyu show [OPTIONS] COMMAND [ARGS]...
```

config

show current cli configuration

```
pyu show config [OPTIONS]
```

group

show group details [Admin only]

```
pyu show group [OPTIONS] NAME
```

Arguments

NAME
Required argument

issue

show issue details

```
pyu show issue [OPTIONS] ID
```

Arguments

ID

Required argument

project

show project details [Admin only]

```
pyu show project [OPTIONS] ID
```

Arguments

ID

Required argument

role

show role details [Admin only]

```
pyu show role [OPTIONS] NAME
```

Arguments

NAME

Required argument

user

show user details [Admin only]

```
pyu show user [OPTIONS] LOGIN
```

Arguments

LOGIN

Required argument

update

update existing youtrack resources

```
pyu update [OPTIONS] COMMAND [ARGS]...
```

group

update a group [Admin only]

```
pyu update group [OPTIONS] NAME
```

Arguments

NAME

Required argument

issue

update an issue

```
pyu update issue [OPTIONS] ID
```

Options

--summary <summary>

--description <description>

--command <command>
command to apply to issue.

--comment <comment>
comment to add to issue

Arguments

ID

Required argument

role

update a role [Admin only]

```
pyu update role [OPTIONS] NAME
```

Options

+permission <add_permissions>
add permission

-permission <remove_permissions>
remove permission

Arguments

NAME
Required argument

user

update a user [Admin only]

```
pyu update user [OPTIONS] LOGIN
```

Options

--name <name>

--email <email>

--password <password>

+group <add_groups>
add user to group

-group <remove_groups>
remove user from group

Arguments

LOGIN
Required argument

p

`pyutrack.connection`, 15

`pyutrack.resources`, 16

Symbols

- +group <add_groups>
 - pyu-update-user command line option, 14, 34
- +permission <add_permissions>
 - pyu-update-role command line option, 14, 34
- auto-join, -no-auto-join
 - pyu-new-group command line option, 9, 29
- base_url <base_url>
 - pyu command line option, 5, 25
- command <command>
 - pyu-new-issue command line option, 9, 29
 - pyu-update-issue command line option, 13, 33
- comment <comment>
 - pyu-update-issue command line option, 13, 33
- debug, -no-debug
 - pyu command line option, 5, 25
- description <description>
 - pyu-new-group command line option, 9, 29
 - pyu-new-issue command line option, 9, 29
 - pyu-new-role command line option, 10, 30
 - pyu-update-issue command line option, 13, 33
- email <email>
 - pyu-update-user command line option, 14, 34
- filter <filter>
 - pyu-list-issues command line option, 7, 27
- force <force>
 - pyu-new-config command line option, 9, 29
- group <group>
 - pyu-list-roles command line option, 8, 28
 - pyu-list-users command line option, 8, 28
 - pyu-new-user command line option, 11, 31
- issue <issue>
 - pyu-list-links command line option, 7, 27
- lead <lead>
 - pyu-new-project command line option, 10, 30
- limit <limit>
 - pyu-list-issues command line option, 7, 27
- name <name>
 - pyu-update-user command line option, 14, 34
- password <password>
 - pyu command line option, 5, 25
 - pyu-update-user command line option, 14, 34
- path <path>
 - pyu-new-config command line option, 9, 29
- permission <permission>
 - pyu-list-users command line option, 8, 28
 - pyu-new-role command line option, 10, 30
- project <project>
 - pyu-list-issues command line option, 7, 27
 - pyu-list-users command line option, 8, 28
- query <query>
 - pyu-list-users command line option, 8, 28
- role <role>
 - pyu-list-permissions command line option, 7, 27
 - pyu-list-users command line option, 8, 28
- summary <summary>
 - pyu-update-issue command line option, 13, 33
- user <user>
 - pyu-list-groups command line option, 7, 27
 - pyu-list-roles command line option, 8, 28
- username <username>
 - pyu command line option, 5, 25
- watch <watch>
 - pyu command line option, 5, 25
- group <remove_groups>
 - pyu-update-user command line option, 14, 34
- permission <remove_permissions>
 - pyu-update-role command line option, 14, 34

A

- api_url (pyutrack.connection.Connection attribute), 15
- assignee (pyutrack.resources.Issue attribute), 16

B

- base_url (pyutrack.connection.Connection attribute), 15

C

- command() (pyutrack.resources.Issue method), 16

Connection (class in pyutrack.connection), 15
 cookies (pyutrack.connection.Credentials attribute), 15
 create() (pyutrack.resources.Group class method), 16
 create() (pyutrack.resources.Issue class method), 16
 create() (pyutrack.resources.IssueLinkType class method), 17
 create() (pyutrack.resources.Project class method), 17
 create() (pyutrack.resources.Role class method), 18
 create() (pyutrack.resources.User class method), 18
 Credentials (class in pyutrack.connection), 15
 credentials (pyutrack.connection.Connection attribute), 15

D

delete() (pyutrack.connection.Connection method), 15
 delete() (pyutrack.resources.Group method), 16
 delete() (pyutrack.resources.Issue method), 16
 delete() (pyutrack.resources.IssueLinkType method), 17
 delete() (pyutrack.resources.Project method), 17
 delete() (pyutrack.resources.Role method), 18
 delete() (pyutrack.resources.User method), 18
 description (pyutrack.resources.Project attribute), 17

E

EMAIL

pyu-new-user command line option, 11, 31

F

fix_auth() (in module pyutrack.connection), 16
 from_keyring() (pyutrack.connection.Credentials static method), 15

G

get() (pyutrack.connection.Connection method), 15
 get() (pyutrack.resources.Group method), 16
 get() (pyutrack.resources.Issue method), 16
 get() (pyutrack.resources.IssueLinkType method), 17
 get() (pyutrack.resources.Project method), 17
 get() (pyutrack.resources.Role method), 18
 get() (pyutrack.resources.User method), 18
 get_groups() (pyutrack.resources.User method), 18
 get_issue_links() (pyutrack.resources.Issue method), 16
 get_issues() (pyutrack.resources.Project method), 17
 get_permissions() (pyutrack.resources.Role method), 18
 get_roles() (pyutrack.resources.Group method), 16
 get_roles() (pyutrack.resources.User method), 18
 Group (class in pyutrack.resources), 16
 groups (pyutrack.resources.User attribute), 18

I

ID

pyu-delete-issue command line option, 6, 26
 pyu-delete-project command line option, 6, 26

pyu-new-project command line option, 10, 30
 pyu-show-issue command line option, 12, 32
 pyu-show-project command line option, 12, 32
 pyu-update-issue command line option, 13, 33

id (pyutrack.resources.Issue attribute), 16
 id (pyutrack.resources.Project attribute), 17
 Issue (class in pyutrack.resources), 16
 issue_links (pyutrack.resources.Issue attribute), 16
 IssueLink (class in pyutrack.resources), 17
 IssueLinkType (class in pyutrack.resources), 17
 issues (pyutrack.resources.Project attribute), 17

K

KEYRING_COOKIE (pyutrack.connection.Credentials attribute), 15
 KEYRING_PASSWORD (pyutrack.connection.Credentials attribute), 15

L

lead (pyutrack.resources.Project attribute), 17
 link (pyutrack.resources.Issue attribute), 16
 list() (pyutrack.resources.Group class method), 16
 list() (pyutrack.resources.Issue class method), 16
 list() (pyutrack.resources.IssueLinkType class method), 17
 list() (pyutrack.resources.Permission class method), 17
 list() (pyutrack.resources.Project class method), 17
 list() (pyutrack.resources.Role class method), 18
 list() (pyutrack.resources.User class method), 18
 load_from_keyring() (pyutrack.connection.Credentials method), 15

LOGIN

pyu-delete-user command line option, 6, 26
 pyu-new-user command line option, 11, 31
 pyu-show-user command line option, 12, 32
 pyu-update-user command line option, 14, 34
 login() (pyutrack.connection.Connection method), 15

N

NAME

pyu-delete-group command line option, 6, 26
 pyu-new-group command line option, 9, 29
 pyu-new-project command line option, 10, 30
 pyu-new-role command line option, 10, 30
 pyu-new-user command line option, 11, 31
 pyu-show-group command line option, 11, 31
 pyu-show-role command line option, 12, 32
 pyu-update-group command line option, 13, 33
 pyu-update-role command line option, 14, 34

P

PASSWORD

pyu-new-user command line option, 11, 31

- password (pyutrack.connection.Credentials attribute), 15
 - Permission (class in pyutrack.resources), 17
 - permissions (pyutrack.resources.Role attribute), 18
 - persist() (pyutrack.connection.Credentials method), 15
 - post() (pyutrack.connection.Connection method), 15
 - priority (pyutrack.resources.Issue attribute), 16
 - PROJECT
 - pyu-new-issue command line option, 10, 30
 - Project (class in pyutrack.resources), 17
 - put() (pyutrack.connection.Connection method), 15
 - pyu command line option
 - base_url <base_url>, 5, 25
 - debug, -no-debug, 5, 25
 - password <password>, 5, 25
 - username <username>, 5, 25
 - watch <watch>, 5, 25
 - pyu-delete-group command line option
 - NAME, 6, 26
 - pyu-delete-issue command line option
 - ID, 6, 26
 - pyu-delete-project command line option
 - ID, 6, 26
 - pyu-delete-user command line option
 - LOGIN, 6, 26
 - pyu-list-groups command line option
 - user <user>, 7, 27
 - pyu-list-issues command line option
 - filter <filter>, 7, 27
 - limit <limit>, 7, 27
 - project <project>, 7, 27
 - pyu-list-links command line option
 - issue <issue>, 7, 27
 - pyu-list-permissions command line option
 - role <role>, 7, 27
 - pyu-list-roles command line option
 - group <group>, 8, 28
 - user <user>, 8, 28
 - pyu-list-users command line option
 - group <group>, 8, 28
 - permission <permission>, 8, 28
 - project <project>, 8, 28
 - query <query>, 8, 28
 - role <role>, 8, 28
 - pyu-new-config command line option
 - force <force>, 9, 29
 - path <path>, 9, 29
 - pyu-new-group command line option
 - auto-join, -no-auto-join, 9, 29
 - description <description>, 9, 29
 - NAME, 9, 29
 - pyu-new-issue command line option
 - command <command>, 9, 29
 - description <description>, 9, 29
 - PROJECT, 10, 30
 - SUMMARY, 10, 30
 - pyu-new-project command line option
 - lead <lead>, 10, 30
 - ID, 10, 30
 - NAME, 10, 30
 - pyu-new-role command line option
 - description <description>, 10, 30
 - permission <permission>, 10, 30
 - NAME, 10, 30
 - pyu-new-user command line option
 - group <group>, 11, 31
 - EMAIL, 11, 31
 - LOGIN, 11, 31
 - NAME, 11, 31
 - PASSWORD, 11, 31
 - pyu-show-group command line option
 - NAME, 11, 31
 - pyu-show-issue command line option
 - ID, 12, 32
 - pyu-show-project command line option
 - ID, 12, 32
 - pyu-show-role command line option
 - NAME, 12, 32
 - pyu-show-user command line option
 - LOGIN, 12, 32
 - pyu-update-group command line option
 - NAME, 13, 33
 - pyu-update-issue command line option
 - command <command>, 13, 33
 - comment <comment>, 13, 33
 - description <description>, 13, 33
 - summary <summary>, 13, 33
 - ID, 13, 33
 - pyu-update-role command line option
 - +permission <add_permissions>, 14, 34
 - permission <remove_permissions>, 14, 34
 - NAME, 14, 34
 - pyu-update-user command line option
 - +group <add_groups>, 14, 34
 - email <email>, 14, 34
 - name <name>, 14, 34
 - password <password>, 14, 34
 - group <remove_groups>, 14, 34
 - LOGIN, 14, 34
 - pyutrack.connection (module), 15
 - pyutrack.resources (module), 16
- ## R
- reporter (pyutrack.resources.Issue attribute), 17
 - reset_cookies() (pyutrack.connection.Credentials method), 15
 - reset_password() (pyutrack.connection.Credentials method), 15
 - Role (class in pyutrack.resources), 17

roles (pyutrack.resources.Group attribute), 16

roles (pyutrack.resources.User attribute), 18

S

status (pyutrack.resources.Issue attribute), 17

SUMMARY

pyu-new-issue command line option, 10, 30

U

update() (pyutrack.resources.Group method), 16

update() (pyutrack.resources.Issue method), 17

update() (pyutrack.resources.IssueLinkType method), 17

update() (pyutrack.resources.Role method), 18

update() (pyutrack.resources.User method), 18

updater (pyutrack.resources.Issue attribute), 17

User (class in pyutrack.resources), 18

username (pyutrack.connection.Credentials attribute), 16