

---

# **Asya Documentation**

***Release 0.0***

**Marek Suchánek**

**Jan 02, 2018**



---

## Contents:

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	How it works . . . . .	1
1.2	MI-PYT . . . . .	1
1.3	Installation . . . . .	1
1.4	Usage . . . . .	2
1.5	License . . . . .	2
<b>2</b>	<b>CLI</b>	<b>3</b>
2.1	asya . . . . .	3
<b>3</b>	<b>API</b>	<b>5</b>
3.1	asya.cli . . . . .	5
3.2	asya.exceptions . . . . .	5
3.3	asya.logic . . . . .	5
3.4	asya.supervisor . . . . .	6
<b>4</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>



# CHAPTER 1

---

## Introduction

---

**AS**ynchronously-**Y**our-**A**cquaintances is simple Python-powered asynchronous client for GitHub API that allows you to list acquaintances from [GitHub](#) issues = all people who commented the issues in which is given user involved (author, commenter, assigned, mentioned). It has of course the ability to filter issues.

### 1.1 How it works

As the name says, it is asynchronous - so all requests are made asynchronously making the tool quite faster than classic synchronous approach. It uses [asyncio](#) and [aiohttp](#). Task done by this tool can be sometimes quite complex and reaches a limits of [GitHub API](#). For extending API rate limit you can provide [personal access token](#) via `--token`. Nevertheless, it may happen that you reach limit even with that - you can use `--wait_rate_limit` flag and application will quietly wait until the limit resets (about 1 hour). Other problem is that [GitHub Search API](#) provides only 1000 results so you need to use filter & sort appropriately in some cases.

### 1.2 MI-PYT

Task for [MI-PYT](#) course (FIT, CTU in Prague, 2017/18) is to implement the core module of this application which is `asya.logic`. Mandatory is to use [asyncio](#), [aiohttp](#) and call appropriate methods on the supervisor object of `asya.supervisor.Asyasupervisor` class.

### 1.3 Installation

You can simply install Asya in standard Python way (system-wide or with virtual env). Python 3.4 or higher is required ([asyncio](#) is in Python since 3.4).

```
python setup.py install
```

## 1.4 Usage

After installation, you can use the CLI of Asya (for more, visit [CLI docs](#)):

```
asya --help
asya MarekSuchanek --token <your-secret-API-token>
```

## 1.5 License

This project is licensed under the MIT License (see the LICENSE file for more details).

## 2.1 asya

Asya Command Line Interface (via [click](#))

```
asya [OPTIONS] USERNAME
```

### Options

- i, --involvement** <involvement>  
How is given user involved in issues.
- text** <text>  
Text to filter issues.
- in** <in>  
In what should be text searched.
- type** <type>  
Restriction to just issues or just PRs.
- state** <state>  
Restriction to just issues or just PRs.
- created** <created>  
Date expression to filter by created date.
- updated** <updated>  
Date expression to filter by updated date.
- label** <label>  
String to filter by label.
- language** <language>  
String to filter by language.

**--sort** <sort>  
Sort search results (important for >1000 results).

**--order** <order>  
Sort order of results (important for >1000 results).

**-t, --token** <token>  
Personal GitHub token.

**-w, --wait-rate-limit**  
Wait for rate limit reset if needed.

**-s, --skip-404**  
Skip not-found GitHub resources (such as disabled repos).

**-b, --progress-bar**  
Toggle progress bar (default false).

**--info, --no-info**  
Toggle info texts (default true).

**--api-endpoint** <api\_endpoint>  
How is given user involved in issues.

**--debug**  
Debug mode (not catching other exceptions).

**--version**  
Show the version and exit.

## Arguments

**USERNAME**  
Required argument

## Environment variables

**GITHUB\_TOKEN**  
Provide a default for `-t`



### 3.1 asya.cli

`asya.cli.create_search_specs` (*username, sort, order, text, involvement, query\_opts*)  
Create GitHub issues search specification (params dict)

`asya.cli.no_print` (*\*args, \*\*kwargs*)  
Dummy method for not actually printing anything

`asya.cli.print_result` (*result*)  
Print Asya result nicely

`asya.cli.setup_info_msgs` (*supervisor*)  
Setup text info printing for given supervisor

`asya.cli.setup_progressbar` (*supervisor*)  
Setup progressbar for given supervisor

### 3.2 asya.exceptions

**exception** `asya.exceptions.AsyException` (*data, headers*)  
Exception of Asya caused during communication with the GitHub API

**Variables**

- **data** (*dict*) – API response data causing the exception
- **headers** (*dict*) – headers of the API response that caused the exception

### 3.3 asya.logic

---

**Important:** This is the module to be implemented!

---

`asya.logic.gather_acquaintances` (*search\_specs*, *supervisor*)

Gather acquaintances from GitHub issues and comments with given *search\_specs*. Return a dict mapping usernames to numbers of comments.

Uses `asyncio` and `aiohttp.ClientSession`.

```
>>> gather_acquaintances({'q': 'author:MarekSuchanek'}, supervisor)
{'MarekSuchanek': 7, 'hroncok': 15, 'encukou': 10}
```

For more information about the *search\_specs* content visit the [GitHub Search API docs](#).

This function may not be called while the `asyncio` event loop is running. It starts and stops the event loop automatically.

#### Parameters

- ***search\_specs*** (*dict*) – dictionary with search specification (params for the search)
- ***supervisor*** (`asya.supervisor.Asyasupervisor`) – supervisor object used for this gathering

**Returns** dictionary with usernames as keys and number of comments as values

**Return type** `dict`

## 3.4 asya.supervisor

**class** `asya.supervisor.Asyasupervisor` (*api\_endpoint*, *token*, *wait\_rate\_limit*, *skip\_404*,  
*per\_page=100*)

Supervisor for running the Asya gathering `async` procedure. It contains some procedure-wide setting and calls.

The `gather_acquaintances` function must call supervisor's `report_` methods at the right places (described in docstrings):

```
...
# receiving and processing the issue
supervisor.report_issue(issue)
# additional work with the issue
...
```

Your implementation may add custom callbacks:

```
def my_procedure(issue):
    ...

supervisor.callbacks['issue'].append(my_procedure)
# my_procedure will be called when report_issue is
# called on the supervisor object
```

You may also use `supervisor.obj` for your data as you need. Do not touch other parameters nor edit this class!

#### Variables

- ***api\_endpoint*** – API endpoint to be used for communication

- **token** – API token to be used (you can use `has_token`)
- **wait\_rate\_limit** – True if the app should wait until the rate limit resets after it is exceeded, False if exceeding the limit should cause *AsyaException*
- **skip\_404** – True the app should skip 404 errors, False if they should raise *AsyaException*
- **per\_page** – size of page for API requests

**has\_token**

True if the API token is set

**obj = None**

User obj (can be anything)

**report\_comment** (*comment*)

Method to be called after processing of a single comment

**Parameters** **issue** (*dict*) – GitHub comment (data from API)

**report\_issue** (*issue*)

Method to be called after processing of a single issue

**Parameters** **issue** (*dict*) – GitHub issue (data from API)

**report\_issues\_search\_page** (*page, number*)

Method to be called before processing of a single result page

**Parameters**

- **issue\_search\_page** (*dict*) – page of issues search (data from API)
- **page\_number** (*int*) – number of the page

**report\_skip** (*headers*)

Method to be called whenever some result is skipped (see `skip_404`)

**Parameters** **headers** (*dict*) – headers of the API response that caused skipping

**report\_wait** (*active, headers*)

Method to be called whenever changing waiting state (see `wait_rate_limit`)

**Parameters**

- **active** (*bool*) – True if wait is starting, False if ending
- **headers** (*dict*) – headers of the API response that caused waiting



## CHAPTER 4

---

### Indices and tables

---

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



### **a**

- `asya.cli`, 5
- `asya.exceptions`, 5
- `asya.logic`, 6
- `asya.supervisor`, 6





## Symbols

- api-endpoint <api\_endpoint>  
    asya command line option, 4
- created <created>  
    asya command line option, 3
- debug  
    asya command line option, 4
- in <in>  
    asya command line option, 3
- info, -no-info  
    asya command line option, 4
- label <label>  
    asya command line option, 3
- language <language>  
    asya command line option, 3
- order <order>  
    asya command line option, 4
- sort <sort>  
    asya command line option, 3
- state <state>  
    asya command line option, 3
- text <text>  
    asya command line option, 3
- type <type>  
    asya command line option, 3
- updated <updated>  
    asya command line option, 3
- version  
    asya command line option, 4
- b, -progress-bar  
    asya command line option, 4
- i, -involvement <involvement>  
    asya command line option, 3
- s, -skip-404  
    asya command line option, 4
- t, -token <token>  
    asya command line option, 4
- w, -wait-rate-limit  
    asya command line option, 4

## A

- asya command line option
  - api-endpoint <api\_endpoint>, 4
  - created <created>, 3
  - debug, 4
  - in <in>, 3
  - info, -no-info, 4
  - label <label>, 3
  - language <language>, 3
  - order <order>, 4
  - sort <sort>, 3
  - state <state>, 3
  - text <text>, 3
  - type <type>, 3
  - updated <updated>, 3
  - version, 4
  - b, -progress-bar, 4
  - i, -involvement <involvement>, 3
  - s, -skip-404, 4
  - t, -token <token>, 4
  - w, -wait-rate-limit, 4
  - USERNAME, 4
- asya.cli (module), 5
- asya.exceptions (module), 5
- asya.logic (module), 6
- asya.supervisor (module), 6
- AsyaException, 5
- AsyaSupervisor (class in asya.supervisor), 6

## C

- create\_search\_specs() (in module asya.cli), 5

## E

- environment variable
  - GITHUB\_TOKEN, 4

## G

- gather\_acquaintances() (in module asya.logic), 6

## H

has\_token (asya.supervisor.AsyaSupervisor attribute), [7](#)

## N

no\_print() (in module asya.cli), [5](#)

## O

obj (asya.supervisor.AsyaSupervisor attribute), [7](#)

## P

print\_result() (in module asya.cli), [5](#)

## R

report\_comment() (asya.supervisor.AsyaSupervisor method), [7](#)

report\_issue() (asya.supervisor.AsyaSupervisor method), [7](#)

report\_issues\_search\_page() (asya.supervisor.AsyaSupervisor method), [7](#)

report\_skip() (asya.supervisor.AsyaSupervisor method), [7](#)

report\_wait() (asya.supervisor.AsyaSupervisor method), [7](#)

## S

setup\_info\_msgs() (in module asya.cli), [5](#)

setup\_progressbar() (in module asya.cli), [5](#)

## U

USERNAME

asya command line option, [4](#)