
Imap-CLI Documentation

Release 0.7

Romain Soufflet (Gentux)

Jun 03, 2018

Contents

1	Installation	3
2	Configuration	5
2.1	Loading default config	5
2.2	Loading config from configuration file	5
2.3	Loading config from python code	6
3	Connection	7
4	Status	9
5	Search	11
6	IMAP Shell	13
6.1	Available commands	13
7	Roadmap	15
7.1	v0.1	15
7.2	v0.2	15
7.3	v0.3	15
7.4	v0.4	15
7.5	v0.5	16
7.6	v0.6	16
7.7	v0.7 (current)	16
8	How to contribute	17
8.1	Submit a patch	17
8.2	If you're not a developer	17
8.3	There are a lot to do	17
9	Indices and tables	19

Everyone is receiving more and more emails days after days and every mail client I know forced users to adapt their working environment.

This is unacceptable, mails are simple. Mail management softwares must adapt to us !

Imap-CLI aims to provide a command line interface, a python API and a REST API to consult and manage IMAP accounts:

- Get IMAP account status (New mails, mail counting... etc...)
- Get list of mails in INBOX (or any other directory)
- Search amongst mails by tag, header, or full text search
- Read mail
- Flag mail (Read, Unread, Delete... etc...)

Contents:

CHAPTER 1

Installation

To install Imap-CLI, you can use pip:

```
pip install imap-cli
```

Or install it from its source code:

```
git clone https://github.com/Gentux/imap-cli.git
cd imap-cli
python setup.py install --user
```

This installation will provide command line tools allowing you to do simple task an IMAP account:

```
* imap-api: Provide a simple REST API to access your mailbox information
* imap-cli-status: List directories and display count of mail *seen*, *unseen* and
↳ *recent*
* imap-cli-list: List mail in specified directory
* imap-cli-search: Search through specified directory for tag or plain text
* imap-cli-read: Display a mail on standard output
* imap-cli-flag: Set or Unset flag on specified set of mail
* imap-shell: A little shell to query mailbox informations
```

Each one of these command has its own *documentation* with the `-help` options

You can also find in Imap-CLI source code 2 bash script. The first one is **imapcli**, it is a little wrapper to help user with mail command:

```
imapcli status
imapcli list
imapcli read 23
```

Is equivalent to:

```
imap-cli-status
imap-cli-list
imap-cli-read 23
```

You can also access to help message of the wanted command with:

```
imapcli help status
```

The last script included in source code is **imapcli_bash_completion.sh**. This script give autocompletion to **imapcli** script. To install it, you just have to do:

```
cp imapcli_bash_completion.sh /etc/bash_completion.d/imapcli
source ~/.bashrc
```

And then, autocompletion will be available.

Imap-CLI need to have information about your IMAP account to connect to it. And to do so, it provide several ways to retrieve those configurations.

2.1 Loading default config

Load an *empty* configuration dictionary with default values listed in file *config-example.ini*:

```
from imap_cli import config
conf = config.new_context()
```

This is usually not what you want, this configuration method is used by unit tests

2.2 Loading config from configuration file

If you intend to use command line tools, this is definitely the best method.

Each command line tools will provide the options *-c*, *-config-file* in order to let you specify your very own config file. It is also helpful to have multiple file in case you have multiple IMAP account.

There is a default configuration file wich will be used if no one is specified, it's *~/.config/imapcli*.

This file can contains the following options:

```
[imap]
hostname = imap.example.org
username = username
password = secret
ssl = True

[display]
format_list =
```

(continues on next page)

(continued from previous page)

```
ID:          {mail_id}
Flags:       {flags}
From:        {from}
To:          {to}
Date:        {date}
Subject:     {subject}
format_thread = {uid} {subject} <<< FROM {from}
format_status = {directory:>20} : {count:>5} Mails - {unseen:>5} Unseen - {recent:>5}
↳Recent
limit = 10
```

2.2.1 SASL Authentication

In addition to the standard *LOGIN* authentication illustrated above, Imap-CLI also supports authentication via SASL. This can be specified in the **[imap]** section of the config:

```
[imap]
hostname = imap.example.org
username = username
sasl_auth = OAUTHBEARER
sasl_ir = abcde12345
ssl = True
```

Here *sasl_auth* is the authentication method and *sasl_ir* is the initial response (or the client response to the first server challenge). If the *sasl_ir* contains non-printable characters, such as the SOH (start of heading) character, you may find it easier to generate the config file programatically.

Additionally, for SASL XOAUTH2 authentication, Imap-CLI can simply take a *bearer_access_token* instead of the *sasl_ir*, like so:

```
[imap]
hostname = imap.example.org
username = username
sasl_auth = XOAUTH2
bearer_access_token = abcde12345
ssl = True
```

Imap-CLI will then automatically construct the SASL XOAUTH2 initial response.

Warning: Don't forget to set correct permission on these files ! It will store your IMAP Account password, allow only your own user to read this file

2.3 Loading config from python code

If you need to connect and retrieve mail information with a python script, you can load any config file

But you can also use a python structure to store your information and load it from a *dict*. In fact, config it's just a dict, the following method will just "complete" your dict.

Every missing key will take the default value.

CHAPTER 3

Connection

Once *configuration* is done, you can connect to your imap account

CHAPTER 4

Status

Status module allow you to get the actual state of your IMAP account. It will tell you for each one of your directories

- The name of directory
- Total number of mail in this directory
- Number of *Recent* mail in this directory
- Number of *Unseen* mail in this directory

This module aim to provide all search options available in IMAP protocols without having to read the entire IMAP RFC.

For now, you can do following searches

- Simple listing of all mails
- Search by tag
- Search by flag
- Search full text in mails body
- search by subject
- search by mail adresse, in *bcc*, *cc*, *from* or *to* header fields
- search by date, *before*, *after* or *on* a particular date
- search by header, given a pair of haeder name and header value to search
- search by size, *larger* or *smaller* than given value

Moreover, a few logical combination of the above

- Not
- Or

The *imap-shell* provide you a minimalist shell to query your IMAP account.

As the other commands, it will use the default configuration file if no other is specified.

6.1 Available commands

- `cd`: Select another mailbox (ex: `cd INBOX`)
- `flag`: Put a flag on a mail
- `help`: Display available command or short description of specified command
- `list`: List mail in selected mailbox
- `quit`: Exit the shell
- `read`: Read specified mail
- `search`: Search through mailbox
- `status`: Display IMAP account statuses

The actual version of Imap-CLI is 0.7.

Impa-CLI aims to map all IMAP protocols functionality within a simple python API, and points in development are listed below

7.1 v0.1

- Status (list directory and new mail per directory)
- List (list content of a directory)
- Read (display content of specified email)

7.2 v0.2

- Basic Search (search amongs tag, subject and full text within mails)
- Tags (Add or Remove tag from mails)

7.3 v0.3

- Rewrite code structure to ease the creation of API
- Documentation

7.4 v0.4

- Advanced Search

- Use UID instead of volatile mails id
- Read all type of mail and handle attachments

7.5 v0.5

- List mail by thread
- Display threads

7.6 v0.6

- Clean up and complete documentation
- Clean up code
- Test coverage > 90%
- Copy, Move and Delete methods

7.7 v0.7 (current)

- Python 3 supported
- Clean up code

All contributions are welcome.

8.1 Submit a patch

To submit a patch

- Step 1: Fork the project on github.
- Step 2: Apply your patch
- Step 3: Ensure all unit test pass, using **tox**
- Step 4: Create a pull request

8.2 If you're not a developer

You can [open new issues](#) for:

- Ask any questions
- Report Bugs
- Request new feature

8.3 There are a lot to do

Feel free to contact any maintainers you can found on github. There always a lot of things to do.

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`