

---

# **python-pop3server Documentation**

***Release 1.0***

**Michael Lang**

**Aug 23, 2017**



---

## Contents

---

<b>1</b>	<b>Indices and tables</b>	<b>3</b>
<b>2</b>	<b>currently implemented:</b>	<b>5</b>
<b>3</b>	<b>known limitations:</b>	<b>7</b>
<b>4</b>	<b>Example usage:</b>	<b>9</b>
4.1	Terminal 1: . . . . .	9
4.2	Terminal 2: . . . . .	10
	<b>Python Module Index</b>	<b>13</b>



Contents:



# CHAPTER 1

---

## Indices and tables

---

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

a python based POP3 Server implementation

due to the lack of pure python based POP3 Servers without mass of dependencies (twisted, ...) I've created this POP3 Protocol based implementation. The initial need for it was to get an application ported which was only able to use POP3 a mechanism. As Microsofts Exchange POP3 implementation isn't really Cluster able the idea to have some kind of proxy/translater as perdition but with protocol change (for example POP3 to IMAP4).

so don't expect to much from this Proof-of-concept as the focus is only to have a single Application transparently talking to an IMAP server even though it's implementation is POP3 only.





## CHAPTER 2

---

currently implemented:

---

- POP3 Server accepting requestes forwarding to one (start up parameter based) IMAP Server
- **POP3 Protocol types:**
  - QUIT
  - STAT
  - LIST
  - RETR
  - DELE
  - NOOP
  - RSET
  - TOP
  - UIDL
  - USER
  - PASS



## CHAPTER 3

---

### known limitations:

---

POP3 is an ASCII only protocol so what ever is stored on the IMAP side is converted to ascii with flag 'ignore'. In principal this shouldn't be a problem as your current implementation isn't able to read UTF8 mails too but I didn't have time to test all possibilities to avoid exceptions if convert doesn't work.

**seen as:** UnicodeDecodeError: 'utf8' codec can't decode byte 0xfc in position 1847: invalid start byte  
fix assumed for 1.1



## CHAPTER 4

---

Example usage:

---

### Terminal 1:

server:

```
$ ./pop3server.py -b IMAP --backend_address=192.168.192.13 --backend_
↪port=143 -p 10110
serving POP3 service at 127.0.0.1:10110
2014-03-16 17:50:03,180 [POP3Server.DEBUG 240]: S: +OK POP3 server ready
2014-03-16 17:50:05,935 [POP3Server.DEBUG 382]: C: USER
2014-03-16 17:50:05,936 [POP3Server.DEBUG 396]: S: +OK
2014-03-16 17:50:09,589 [POP3Server.DEBUG 382]: C: PASS
2014-03-16 17:50:09,589 [POP3Server.DEBUG 338]: trying to acquire Lock for_
↪michi
2014-03-16 17:50:09,590 [POP3Server.DEBUG 342]: stating imap connection
2014-03-16 17:50:09,590 [POP3Server.DEBUG 66]: IMAP: connecting to 192.168.
↪192.13:143
2014-03-16 17:50:12,281 [POP3Server.DEBUG 396]: S: +OK maildrop locked and_
↪ready
2014-03-16 17:50:15,259 [POP3Server.DEBUG 382]: C: STAT
2014-03-16 17:50:15,278 [POP3Server.DEBUG 99]: IMAP: fetching INBOX ['2']
2014-03-16 17:50:15,279 [POP3Server.DEBUG 106]: IMAP: retrieve 1
2014-03-16 17:50:15,348 [POP3Server.DEBUG 392]: S: +OK 1 81679
2014-03-16 17:50:22,523 [POP3Server.DEBUG 382]: C: TOP
2014-03-16 17:50:22,530 [POP3Server.DEBUG 388]: S: +OK
2014-03-16 17:53:47,893 [POP3Server.DEBUG 382]: C: QUIT
2014-03-16 17:53:47,894 [POP3Server.DEBUG 137]: IMAP: expunge
2014-03-16 17:53:47,910 [POP3Server.DEBUG 392]: S: +OK POP3 server signing_
↪off
```

## Terminal 2:

client:

```
$ telnet localhost 10110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK POP3 server ready
USER michi
+OK
PASS xxxxxxxx
STAT
+OK maildrop locked and ready
TOP 1 10
+OK
...
Header + 10 Lines Body supressed ...
QUIT
+OK POP3 server signing off
```

### **class** pop3server.**MailboxLocker**

POP3 Mailboxes are single connected, meaning if a user already authenticated successful all others using the same credentials are not allowed to access the same spool. (this would be possible through the backend but breaks POP3 protocol).

**acquire** ()  
inline function to unique locking through all the threads

**acquire\_mailbox** (*name=None*)  
retrieve a lock for a specific mailbox name=xx

**is\_locked** (*name=None*)  
return lock status of name=xx

**release** ()  
inline function to unique locking through all the threads

**release\_mailbox** (*name=None*)  
release the lock for a specific mailbox name=xx

### **class** pop3server.**POP3Backend** (*protocol=None*)

POP3 Backend Interface Class this class defines the methods your Backend implementation should provide

**authenticate** (*username=None, password=None*)  
authenticate the user with given credentials

**cleanup** ()  
cleanup the POP3Backend maildrop for the user (eq. remove files, delete tables, expunge ...)

**delete** (*num=None*)  
remove given item identifier fro the POP3Backend maildrop

**destroy** ()  
close handles for POP3Backend

**fetch** ()  
populate self.protocol.messages with what ever is in the POP3Backend maildrop for the user

**revert** ()  
revert already marked items for deletion/rollback transactions, ...

```

class pop3server.POP3Backend_IMAP (protocol=None, host=None, port=143, timeout=5.0)
    IMAP based Backend this backend provides the “translation” between POP3 and IMAP details as follows:
        •IMAP message -> converted -> ascii (flag ignore=just leave the character out of the Unicode result)
        •IMAP INBOX messages only (if you filter move messages to subfolders they are not recognized by the Backend)
        •POP3 -> plain text authentication only (as this is proxied to the remote)
        •IMAP special user monitor, the user monitor with pwd monitor is used for unittesting and monitoring purpose

    authenticate (username=None, password=None)
        try to proxy authenticate against IMAP with POP3 given values

    cleanup ()
        purge all marked message from IMAP spool

    delete (num=None)
        delete (==flag) message (num=xx) from IMAP spool

    fetch ()
        retrieve all messages from IMAP spool limited to 10 for testing

    revert ()
        revert deleted (==flagged) message from IMAP spool

class pop3server.POP3Backend_IMAPS (protocol=None, host=None, port=993, timeout=5.0)
    IMAP based Backend using SSL as transport currently no certificate handling is done, silently ignored

class pop3server.POP3Message (content=None)
    represents a POP3 Message to be displayed on the User MUA

    as_string ()
        return the complete message

    get_body ()
        return the body of the message

    get_headers ()
        return only the headers of the messages

    unique_id ()
        return a unique ID (based upon POP3 impl. md5 sum of message content)

class pop3server.POP3ServerProtocol (request, client_address, server)
    the POP3 Server protocol implementation http://tools.ietf.org/html/rfc1081 and http://www.ietf.org/rfc/rfc1939.txt

    APOP (digest=None)
        apop command An alternate method of authentication is required which provides for both origin authentication and replay protection, but which does not involve sending a password in the clear over the network.
        NOT implemented

    AUTHORIZATION ()
        the authorization or welcome banner

    DELE (msg=None)
        dele command removes a specific message from the spool

```

**LIST** (*msg=None*)

list command returns either one specific or without arguments a list of all messages enumerated and size  
Examples:

```
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
S: ...
C: LIST 2
S: +OK 2 200
```

**NOOP** ()

noop command for idle connections to avoid tcp timeouts on firewalls or similar

**PASS** (*credentials=None*)

pass command sets the password credentials part

**QUIT** ()

quit transaction and session command

**RETR** (*msg=None*)

retr command returns a complete specific message

**RSET** ()

rset command resets all actions taken by the MUA (delete messages is reverted)

**STAT** ()

stat command returns count and size of messages to the MUA

**TOP** (*msg=None, lines=None*)

top command returns from a specific message all headers plus n lines of the body

**UIDL** (*msg=None*)

uidl command returns a spool unique message id based upon the md5 hashdigest of the message

**USER** (*name=None*)

user command sets the user credentials part

**handle** ()

core routing to handle the MUA command requests



**p**

pop3server, [10](#)



## A

acquire() (pop3server.MailboxLocker method), 10  
acquire\_mailbox() (pop3server.MailboxLocker method), 10  
APOP() (pop3server.POP3ServerProtocol method), 11  
as\_string() (pop3server.POP3Message method), 11  
authenticate() (pop3server.POP3Backend method), 10  
authenticate() (pop3server.POP3Backend\_IMAP method), 11  
AUTHORIZATION() (pop3server.POP3ServerProtocol method), 11

## C

cleanup() (pop3server.POP3Backend method), 10  
cleanup() (pop3server.POP3Backend\_IMAP method), 11

## D

DELE() (pop3server.POP3ServerProtocol method), 11  
delete() (pop3server.POP3Backend method), 10  
delete() (pop3server.POP3Backend\_IMAP method), 11  
destroy() (pop3server.POP3Backend method), 10

## F

fetch() (pop3server.POP3Backend method), 10  
fetch() (pop3server.POP3Backend\_IMAP method), 11

## G

get\_body() (pop3server.POP3Message method), 11  
get\_headers() (pop3server.POP3Message method), 11

## H

handle() (pop3server.POP3ServerProtocol method), 12

## I

is\_locked() (pop3server.MailboxLocker method), 10

## L

LIST() (pop3server.POP3ServerProtocol method), 11

## M

MailboxLocker (class in pop3server), 10

## N

NOOP() (pop3server.POP3ServerProtocol method), 12

## P

PASS() (pop3server.POP3ServerProtocol method), 12  
POP3Backend (class in pop3server), 10  
POP3Backend\_IMAP (class in pop3server), 11  
POP3Backend\_IMAPS (class in pop3server), 11  
POP3Message (class in pop3server), 11  
pop3server (module), 10  
POP3ServerProtocol (class in pop3server), 11

## Q

QUIT() (pop3server.POP3ServerProtocol method), 12

## R

release() (pop3server.MailboxLocker method), 10  
release\_mailbox() (pop3server.MailboxLocker method), 10  
RETR() (pop3server.POP3ServerProtocol method), 12  
revert() (pop3server.POP3Backend method), 10  
revert() (pop3server.POP3Backend\_IMAP method), 11  
RSET() (pop3server.POP3ServerProtocol method), 12

## S

STAT() (pop3server.POP3ServerProtocol method), 12

## T

TOP() (pop3server.POP3ServerProtocol method), 12

## U

UIDL() (pop3server.POP3ServerProtocol method), 12  
unique\_id() (pop3server.POP3Message method), 11  
USER() (pop3server.POP3ServerProtocol method), 12