

aiosmtpd

An asyncio-based SMTP server

Pycon 2017 Portland, Oregon

May 21, 2017

Barry Warsaw

Every program attempts to expand until it can read mail. Those programs which cannot so expand are replaced by ones which can.

- jwz (though maybe not)

Zawinski's Law of Software Envelopment

Simple Mail Transport Protocol

(and its little cousin **LMTP**)

Local **M**ail **T**ransport **P**rotocol)

Motivation

Why an SMTP server?

- Testing email clients
- Pythonically configurable server
- Experimental protocols
- Production quality server

SMTP and LMTP

Some relevant RFCs

- RFC 821 (1982) - Original SMTP standard
- RFC 5321 (2008) - Defines ESMTP
- RFC 2033 - Defines LMTP
- RFC 1870 - message sizes
- RFC 6531 - internationalization
- (and so on...!)

SMTP in a nutshell

```
220 subdivisions Python SMTP 1.0
HELO limelight
250 subdivisions
MAIL FROM:<geddy@example.com>
250 OK
RCPT TO:<alex@example.com>
250 OK
RCPT TO:<neil@example.com>
250 OK
DATA
354 End data with <CR><LF>.<CR><LF>
```

```
From: Geddy <geddy@example.com>
To: Alex <alex@example.com>
Bcc: Neil <neil@example.com>
Subject: New Music

Hey, we need to record a new album!
```

```
.
250 OK
QUIT
221 Bye
```

RFC 5322

RFC 5321

Variations

Original SMTP

`HELO limelight`

Extended SMTP (ESMTP)

`EHLO limelight`

Local Mail Transport Protocol (LMTP)

`LHLO limelight`

(Very) brief history

- `asynchat/asyncore`
- `smtpd.py` (2001, Python 2.1 a2)
- `lazr.smtptest` (2009)
- `asyncio` (2014, Python 3.4)
- `aiosmtp` (Benjamin Bader)
- `aiosmtpd` (April 2015 - Washington DC)

Moving pictures

Components

- SMTP/LMTP classes - protocol
- Session & envelope - state
- Handlers - events
- Controller - optional, but helpful!

Session

Newly created on every client connection

- socket peer
- SSL details
- ESMTP flag
- event loop

Envelope

HELO/RSET state

- MAIL FROM address
- RCPT TO addresses
- Original content (always bytes)
- Content (bytes or str)
- Additional ESMTP options

SMTP class

- Verbs as coroutines
- STARTTLS
- UTF-8 handling
- Calls the event handler
- Exception hooks
- Extensible via subclassing

SMTP Verbs

HELO

```
async def smtp_HELO(self, hostname):
    if not hostname:
        await self.push('501 Syntax: HELO hostname')
        return
    self._set_rset_state()
    status = await self._call_handler_hook('HELO', hostname)
    if status is MISSING:
        self.session.host_name = hostname
        status = '250 {}'.format(self.hostname)
    await self.push(status)
```

Get the server's time

```
from aiosmtpd.smtp import SMTP
from datetime import datetime

class MySMTPish(SMTP):
    async def smtp_NOW(self, arg):
        if arg == 'UTC':
            now = datetime.utcnow()
        elif not arg:
            now = datetime.now()
        else:
            await self.push('501 Syntax: NOW [UTC]')
        now = now.replace(microsecond=0)
        await self.push('250 {}'.format(now))
```

Get the server's time

```
% telnet localhost 8025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 presto Python SMTP 1.0
NOW
250 2017-05-16 12:05:05
NOW
250 2017-05-16 12:05:08
NOW UTC
250 2017-05-16 19:05:10
NOW AND THEN
501 Syntax: NOW [UTC]
QUIT
221 Bye
```

Event Handlers

Handlers

```
async def handle_VERB(self, server, session, envelope)
```

- Introspection for verbs
- Coroutines
- Return status code
- No default behavior

HELO counter

```
class Counter:
    helo_counter = 0

    async def handle_HELO(self, server, session, envelope, hostname):
        self.session.hostname = hostname
        self.helo_counter += 1
        return '250 OK'

smtp = SMTP(Counter())
run_server_for_a_while(smtp)
print('We saw {} HELOs'.format(smtp.event_handler.helo_counter))
```

Controller

- Runs server in subthread
- Start/stop semantics
- Pass in hostname, port, etc.
- Override *factory()* to use custom SMTP subclasses

Command line


```
$ python3 -m aiosmtpd -n -c MyHandler arg1 arg2
```

```
$ telnet localhost 8025
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.  
220 subdivisions Python SMTP 1.0
```

```
QUIT
```

```
221 Bye
```

```
Connection closed by foreign host.
```

CLI arguments

```
class MyHandler:  
    @classmethod  
    def from_cli(cls, parser, *args):  
        kws = convert(args)  
        return cls(**kws)
```

Acknowledgments

- Benjamin Bader - aiosmtp (no 'd')
- DC Hackathon Crew:
 - Jason Coombs
 - Andrew Kuchling
 - Eric V. Smith
 - Barry Warsaw
 - R. David Murray (honorary)
- Konstantin Volkov
- Matthias Rav
- (and others... thanks!)

aiosmtpd

<https://github.com/aio-libs/aiosmtpd>

<http://aiosmtpd.readthedocs.io/>

Requires at least Python 3.4 (but see GH#16)

Barry Warsaw

barry@python.org

barry@ubuntu.com

[@pumpichank](#)

github.com/warsaw

gitlab.com/warsaw