
Drymail Documentation

Sumit Ghosh

Jan 07, 2021

Contents

1	drymail	1
1.1	Features	1
1.2	Installation	2
1.3	Documentation	2
1.4	Agenda	2
1.5	Contribute	2
1.6	License	2
1.7	The API Documentation	2
1.8	Indices and tables	6
	Python Module Index	7
	Index	9

CHAPTER 1

drymail

Makes sending emails easy and **DRY** — For Python 3.

Drymail is a minimalist wrapper over Python's existing `smtplib` and `email` libraries, designed to be friendly but unrestrictive. Here's how you might send a simple email with an attachment using *drymail*.

```
from drymail import SMTPMailer, Message

client = SMTPMailer(host='smtp.email.com', user='johndoe', password='password',
    ↪tls=True)
message = Message(subject='Congrats on the new job!', sender=('John Doe', 'john@email.
    ↪com'),
                receivers=[('Jane Doe', 'jane@message.com'), 'jane.doe@mail.io'],
    ↪text='When is the party? ;)')
message.attach(filename='/path/to/congrats.pdf', mimetype='application/pdf')

client.send(message)
```

1.1 Features

- Supports creating email with HTML content, plaintext content, or both!
- Supports mentioning contacts in the "John Doe" <john@email.com> format.
- Support standard headers like CC, BCC, Reply-To and Authors.
- Supports injecting custom headers.
- Supports adding attachments.
- And most importantly — the library being minimalist, it doesn't restrict you in any way like some of the most fancier email frameworks do.

1.2 Installation

Install `drymail` by running —

```
$ pip3 install drymail
```

1.3 Documentation

Documentation is available at <https://drymail.readthedocs.io/>

1.4 Agenda

- Test suite.
- Restructure the `Message` class to handle dynamic `prepare`.

1.5 Contribute

All kinds of contribution are welcome.

- Issue Tracker — <https://github.com/drymail/issues>
- Source Code — <https://github.com/drymail>

1.6 License

This project is licensed under the MIT license.

1.7 The API Documentation

If you are looking for information on a specific function, class, or method, this part of the documentation is for you.

1.7.1 `drymail` module

```
class drymail.Message(sender, receivers, subject=None, authors=None, cc=None, bcc=None,  
                     reply_to=None, headers=None, text=None, html=None, pre-  
                     pared_message=None)
```

Bases: `object`

Class representing an email message.

Parameters

- **sender** (*str* or (*str*, *str*)) – The address of the sender. Can be either the email address or a tuple of the name and the email address.
- **receivers** (*list of (str or (str, str))*) – The list of receivers // recipients. Each element can be either an email address or a tuple of a name and an email address.

- **subject** (*str*, *optional*) – The subject of the email
- **authors** (*list of (str or (str, str)), optional*) – The list of authors, to be mentioned in the *Authors* header. Each element can be either an email address or a tuple of a name and an email address.
- **cc** (*list of (str or (str, str)), optional*) – The list of addresses to CC to. Each element can be either an email address or a tuple of a name and an email address.
- **bcc** (*list of (str or (str, str)), optional*) – The list of addresses to BCC to. Each element can be either an email address or a tuple of a name and an email address.
- **reply_to** (*list of (str or (str, str)), optional*) – The list of addresses to mention in the *Reply-To* header. Each element can be either an email address or a tuple of a name and an email address.
- **headers** (*dict*, *optional*) – Custom headers as key-value pairs, to be injected into the email.
- **text** (*str*, *optional*) – The body of the message, as plaintext. At least one among *text* and *html* must be provided.
- **html** (*str*, *optional*) – The body of the message, as HTML. At least one among *text* and *html* must be provided.
- **prepared_message** (*bytes*, *optional*) – A prepared email as bytes. If this is provided, all the other optional parameters will be ignored.

message

The prepared message object.

Type *email.message.Message* object or *email.mime.multipart.MIMEMultipart* object

prepared

Whether the message is prepared, in other words whether *self.message* is available and proper.

Type bool

sender

The address of the sender. Can be either the email address or a tuple of the name and the email address.

Type str or (str, str)

receivers

The list of receivers // recipients. Each element can be either an email address or a tuple of a name and an email address.

Type list of (str or (str, str))

subject

The subject of the email

Type str

authors

The list of authors, to be mentioned in the *Authors* header. Each element can be either an email address or a tuple of a name and an email address.

Type list of (str or (str, str))

cc

The list of addresses to CC to. Each element can be either an email address or a tuple of a name and an email address.

Type list of (str or (str, str))

bcc

The list of addresses to BCC to. Each element can be either an email address or a tuple of a name and an email address.

Type list of (str or (str, str))

reply_to

The list of addresses to mention in the *Reply-To* header. Each element can be either an email address or a tuple of a name and an email address.

Type list of (str or (str, str))

headers

Custom headers as key-value pairs, to be injected into the email.

Type dict

text

The body of the message, as plaintext.

Type str

html

The body of the message, as HTML.

Type str

prepared_message

A prepared email as bytes.

Type bytes

attach (*filename*, *data=None*, *mimetype=None*)

Add a file as attachment to the email.

Parameters

- **filename** (*str*) –
If data is provided: The filename of the file to be attached.
If data not provided: The full name (path + filename) of the file to be attached.
- **data** (*bytes*, *optional*) – The raw content of the file to be attached.
- **mimetype** (*str*, *optional*) – The MIMEType of the file to be attached.

attachments

All the attachments attached to the message.

Returns The filenames of the attachments attached.

Return type list of str

prepare ()

Prepare the *self.message* object.

class drymail.**SMTPMailer** (*host*, *port=None*, *user=None*, *password=None*, *ssl=False*, *tls=False*,
 ***kwargs*)

Bases: object

Wrapper around *smtplib.SMTP* class, for managing a SMTP client.

Parameters

- **host** (*str*) – The hostname of the SMTP server to connect to.

- **port** (*int, optional*) – The port number of the SMTP server to connect to.
- **user** (*str, optional*) – The username to be used for authentication to the SMTP server.
- **password** (*str, optional*) – The password to be used for authentication to the SMTP server.
- **ssl** (*bool, optional*) – Whether to use SSL for the SMTP connection.
- **tls** (*bool, optional*) – Whether to use TLS // *starttls* for the SMTP connection.
- **keyfile** (*str, optional*) – File containing the SSL private key.
- **certfile** (*str, optional*) – File containing the SSL certificate in PEM format.
- **context** (*ssl.SSLContext object*) – The SSL context to be used in the SSL connection.

client

The SMTP client that'd be used to send emails.

Type *smtplib.SMTP* object

connected

Whether there is an active SMTP connection.

Type *bool*

host

The hostname of the SMTP server to connect to.

Type *str*

port

The port number of the SMTP server to connect to.

Type *int*

user

The username to be used for authentication to the SMTP server.

Type *str*

password

The password to be used for authentication to the SMTP server.

Type *str*

ssl

Whether to use SSL for the SMTP connection.

Type *bool*

tls

Whether to use TLS // *starttls* for the SMTP connection.

Type *bool*

close ()

Close the SMTP connection and *quit* the *self.client* object.

connect ()

Create the SMTP connection.

send (message, sender=None, receivers=None)

Send an email through this SMTP client.

Parameters

- **message** (*drymail.Message* object) – The message to be sent.
- **sender** (*str*, *optional*) – The email address of the sender.
- **receivers** (*list of str*, *optional*) – The email addresses of the receivers // recipients.

`drymail.stringify_address(address)`

Converts an address into a string in the “*John Doe*” <*john@example.com*>” format, which can be directly used in the headers of an email.

Parameters **address** (*str* or (*str*, *str*)) – An address. Can be either the email address or a tuple of the name and the email address.

Returns Address as a single string, in the “*John Doe*” <*john@example.com*>” format. Returns *address* unchanged if it’s a single string.

Return type `str`

`drymail.stringify_addresses(addresses)`

Converts a list of addresses into a string in the “*John Doe*” <*john@example.com*>, “*Jane*” <*jane@example.com*>” format, which can be directly used in the headers of an email.

Parameters **addresses** ((*str* or (*str*, *str*)) or *list of (str or (str, str))*) – A single address or a list of addresses which is to be converted into a single string. Each element can be either an email address or a tuple of a name and an email address.

Returns The address(es) as a single string which can be directly used in the headers of an email.

Return type `str`

1.8 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

d

drymail, [2](#)

A

`attach()` (*drymail.Message method*), 4
`attachments` (*drymail.Message attribute*), 4
`authors` (*drymail.Message attribute*), 3

B

`bcc` (*drymail.Message attribute*), 4

C

`cc` (*drymail.Message attribute*), 3
`client` (*drymail.SMTPMailer attribute*), 5
`close()` (*drymail.SMTPMailer method*), 5
`connect()` (*drymail.SMTPMailer method*), 5
`connected` (*drymail.SMTPMailer attribute*), 5

D

`drymail` (*module*), 2

H

`headers` (*drymail.Message attribute*), 4
`host` (*drymail.SMTPMailer attribute*), 5
`html` (*drymail.Message attribute*), 4

M

`Message` (*class in drymail*), 2
`message` (*drymail.Message attribute*), 3

P

`password` (*drymail.SMTPMailer attribute*), 5
`port` (*drymail.SMTPMailer attribute*), 5
`prepare()` (*drymail.Message method*), 4
`prepared` (*drymail.Message attribute*), 3
`prepared_message` (*drymail.Message attribute*), 4

R

`receivers` (*drymail.Message attribute*), 3
`reply_to` (*drymail.Message attribute*), 4

S

`send()` (*drymail.SMTPMailer method*), 5
`sender` (*drymail.Message attribute*), 3
`SMTPMailer` (*class in drymail*), 4
`ssl` (*drymail.SMTPMailer attribute*), 5
`stringify_address()` (*in module drymail*), 6
`stringify_addresses()` (*in module drymail*), 6
`subject` (*drymail.Message attribute*), 3

T

`text` (*drymail.Message attribute*), 4
`tls` (*drymail.SMTPMailer attribute*), 5

U

`user` (*drymail.SMTPMailer attribute*), 5