
Pystmark Documentation

Release 0.4.8

Steve Leonard

Sep 26, 2019

Contents

1	Installation	3
2	Web Framework Integration	5
3	Example Usage	7
3.1	Sending Email	7
3.2	Bounce Handling	10
4	API Reference	13
4.1	Simple API	13
4.2	Advanced API	15
4.3	Message Object	20
4.4	Response Objects	22
4.5	Response Data Wrappers	23
4.6	Exceptions	24
4.7	Base Classes	25
	Python Module Index	27
	Index	29

Pystmark is a wrapper around the Postmark API. It is built on top of [Requests](#).

CHAPTER 1

Installation

Install the extension with one of the following commands:

```
$ easy_install pystmark
```

or alternatively if you have pip installed:

```
$ pip install pystmark
```

Additionally, install [Requests](#) in the same manner if you have not.

CHAPTER 2

Web Framework Integration

- Flask-Pystmark

3.1 Sending Email

See also:

Postmark Send API

See also:

Sending Mail with the Simple API

3.1.1 Sending a single message

To send a single message, create a *Message* object, and pass it to *send()*.

```
import pystmark

API_KEY = 'my_api_key'
SENDER = 'me@example.com'

# Send a single message
message = pystmark.Message(sender=SENDER, to='you@example.com',
                           subject='Hi', text='A message',
                           tag='greeting')

pystmark.send(message, api_key=API_KEY)
```

You can also pass in a dictionary. It will construct the *Message* for you.

```
import pystmark

pystmark.send(dict(sender=SENDER, to='you@example.com', subject='Hi',
                  text='A message', tag='greeting'), api_key=API_KEY)
```

3.1.2 Sending a single message with template

To send a single message with template, create a *Message* object, and pass it to `send_with_template()`. The `template_id` can be found with template meta data.

```
t_model = {'product_name': "Awesome Product",
          'name': "Customer Name",
          'action_url': "http://www.example.net/confirmation/sjdf092hsdf",
          'sender_name': "Product Team",
          'product_address_line1': "Dover",
          'product_address_line2': "DE 19012"}

message = pystmark.Message(sender=PM_SENDER, to='you@example.com',
                          template_id=11111,
                          template_model=t_model,
                          tag='welcome')

pystmark.send_with_template(message, api_key=PM_API_KEY)
```

3.1.3 Sending batched messages

This sends multiple messages in a single http request to Postmark's batch send API. There is a hard limit of 500 messages.

If you want to send the same message but to multiple recipients, you can use `send()`, and construct the message with multiple *to*, *cc* or *bcc* addresses. See *Multiple Recipients*.

```
from pystmark import Message, send_batch

# Send multiple messages (in one batched http request)
recipients = ['you{0}@example.com'.format(i) for i in xrange(20)]
messages = [Message(sender=SENDER, to=to, subject='Hi', text='A message',
                   tag='greeting') for to in recipients]

response = send_batch(messages, api_key=API_KEY)
```

3.1.4 Multiple recipients

The Postmark API allows you to have multiple *to* recipients. The total number of recipients, including *to*, *cc*, and *bcc* is limited to 20.

```
from pystmark import Message, send

message = Message(sender=SENDER, subject='Hi', text='A message',
                 to=['you@example.com', 'him@example.com'],
                 cc=['someone@example.com', 'her@example.com'],
                 bcc='user@example.com')

send(message, api_key=API_KEY)
```

3.1.5 Sender Configuration

You can set defaults for your message sending using the *Advanced API*. For every method in the *Simple API*, there is a corresponding configurable sender object in the *Advanced API*.

```

from pystmark import Message, Sender

default_message = Message(sender=SENDER,
                           subject='Hi',
                           text='Welcome to the site',
                           html='<h1>Welcome to the site</h1>',
                           tag='greeting')

sender = Sender(message=default_message, api_key=API_KEY)

sender.send(dict(to='you@example.com'))

```

3.1.6 Attachments

Attachments are allowed, up to 10MB in size. The attachment sizes are not checked to be under the limit. If you think you might go over the limit, make sure to check yourself. Only certain file extensions are allowed.

```

import pystmark

filename = '/tmp/example.txt'
with open(filename, 'w') as f:
    f.write('demo\n')

message = pystmark.Message(sender='me@example.com',
                           to='you@example.com',
                           text='hi')

# Attach using filename
message.attach_file(filename)

# Attach using binary
with open(filename) as f:
    message.attach_binary(f.read(), filename)

pystmark.send(message, api_key='the key')

```

3.1.7 Email Headers

Custom headers can be added for your email.

```

import pystmark

message = pystmark.Message(sender='me@example.com',
                           to='you@example.com',
                           text='hi')

message.add_header('X-my-custom-header', 'foo')

pystmark.send(message, api_key='the key')

```

3.1.8 Response Errors

Some HTTP status codes will raise a custom Exception. See `Response.raise_for_status()`.

```
from pystmark import send, UnauthorizedError

r = send(dict(sender='me@example.com', to='you@example.com', text='hi'),
         api_key='bad key')

try:
    r.raise_for_status()
except UnauthorizedError:
    print 'Use your real API key'
```

3.1.9 Requests.request Arguments

If you need to pass some arguments to `requests.request()`, you can do so. However, you cannot modify the `data` keyword. It will be ignored if you give it.

```
from pystmark import send, Message

message = Message(sender='me@example.com', to='you@example.com', text='hi')

send(message, api_key='my key', **dict(headers={'X-Something': 'foo'}))
```

3.2 Bounce Handling

See also:

Postmark Bounce API

See also:

Bounce Handling with the Simple API

3.2.1 Retrieving bounced emails

Bounced emails are retrieved with `get_bounces()`. The request must be paginated with the `count` and `offset`. They will default to 25 and 0, respectively. If you provide a `message_id` (saved from the response of a previously sent message), you do not need to provide `count` or `offset`. You can filter bounces by a string match or bounce type.

```
from pystmark import send, get_bounces

API_KEY = 'my key'

# Get all bounces. If we do not paginate, 25 results will be returned at
# offset 0.
get_bounces(count=100, offset=0, api_key=API_KEY)

# Get bounces of a specific type
get_bounces(bounce_type='HardBounce', api_key=API_KEY)

# Get bounces filtered by email string
get_bounces(email_filter='@gmail.com', api_key=API_KEY)

# Get bounces for a message
r = send(dict(sender='me@example.com', to='you@example.com', text='hi'),
```

(continues on next page)

(continued from previous page)

```
        api_key=API_KEY)
get_bounces(message_id=r.message.id, api_key=API_KEY)
```

3.2.2 Retrieving a single bounce

Data for a single bounce can be retrieved given a *bounce_id*.

```
from pystmark import get_bounce, get_bounces

r = get_bounces(api_key='my key')
for bounce in r.bounces:
    get_bounce(bounce.id, api_key='my key')
```

3.2.3 Retrieving the raw dump for a single bounce

The raw email dump can be retrieved with a *bounce_id* or with a *BouncedMessage*.

```
from pystmark import get_bounces, get_bounce_dump

r = get_bounces(api_key='my key')
for bounce in r.bounces:
    # Get dump via BouncedMessage.
    dump = bounce.dump(api_key='my key')
    # Get dump with the simple API
    dump = get_bounce_dump(bounce.id, api_key='my key')
```

3.2.4 Activating a bounced message (re-sending it)

Bounces can be re-sent with activation. Keep in mind that some bounces such as hard bounces should be assumed dead.

```
from pystmark import get_bounces, activate_bounce

r = get_bounces(api_key='my key')
for bounce in r.bounces:
    activate_bounce(bounce.id)
```

3.2.5 Retrieving tags for bounced messages

You can get a list of tags that have bounced messages. Tags are set on the message by you, when they are sent.

```
from pystmark import get_bounces, get_bounce_tags

r = get_bounces(api_key='my key')
for bounce in r.bounces:
    get_bounce_tags(bounce.id)
```

3.2.6 Retrieving delivery statistics

Delivery stats summarize your bounces.

```
from pystmark import get_delivery_stats

r = get_delivery_stats(api_key='my key')
print 'Inactive Messages:', r.inactive
print 'Total bounces:', r.total
print 'Bounces:'
for bounce in r.bounces.values():
    print '\tType:', bounce.type
    print '\t\tName:', bounce.name
    print '\t\tCount:', bounce.count
```


4.1 Simple API

4.1.1 Sending Email

`pystmark.send` (*message*, *api_key=None*, *secure=None*, *test=None*, ***request_args*)
Send a message.

Parameters

- **message** (*dict* or *Message*) – Message to send.
- **api_key** – Your Postmark API key. Required, if *test* is not *True*.
- **secure** – Use the https scheme for the Postmark API. Defaults to *True*
- **test** – Use the Postmark Test API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *SendResponse*

`pystmark.send_batch` (*messages*, *api_key=None*, *secure=None*, *test=None*, ***request_args*)
Send a batch of messages.

Parameters

- **messages** – Messages to send.
- **api_key** – Your Postmark API key. Required, if *test* is not *True*.
- **secure** – Use the https scheme for the Postmark API. Defaults to *True*
- **test** – Use the Postmark Test API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *BatchSendResponse*

4.1.2 Bounce Handling

`pystmark.get_bounces` (*api_key=None, secure=None, test=None, **request_args*)
Get a paginated list of bounces.

Parameters

- **api_key** – Your Postmark API key. Required, if *test* is not *True*.
- **secure** – Use the https scheme for the Postmark API. Defaults to *True*
- **test** – Use the Postmark Test API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *BouncesResponse*

`pystmark.get_bounce` (*bounce_id, api_key=None, secure=None, test=None, **request_args*)
Get a single bounce.

Parameters

- **bounce_id** – The bounce’s id. Get the id with `get_bounces()`.
- **api_key** – Your Postmark API key. Required, if *test* is not *True*.
- **secure** – Use the https scheme for the Postmark API. Defaults to *True*
- **test** – Use the Postmark Test API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *BounceResponse*

`pystmark.get_bounce_dump` (*bounce_id, api_key=None, secure=None, test=None, **request_args*)
Get the raw email dump for a single bounce.

Parameters

- **bounce_id** – The bounce’s id. Get the id with `get_bounces()`.
- **api_key** – Your Postmark API key. Required, if *test* is not *True*.
- **secure** – Use the https scheme for the Postmark API. Defaults to *True*
- **test** – Use the Postmark Test API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *BounceDumpResponse*

`pystmark.activate_bounce` (*bounce_id, api_key=None, secure=None, test=None, **request_args*)
Activate a deactivated bounce.

Parameters

- **bounce_id** – The bounce’s id. Get the id with `get_bounces()`.
- **api_key** – Your Postmark API key. Required, if *test* is not *True*.
- **secure** – Use the https scheme for the Postmark API. Defaults to *True*
- **test** – Use the Postmark Test API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *BounceActivateResponse*

`pystmark.get_bounce_tags` (*api_key=None, secure=None, test=None, **request_args*)

Get a list of tags for bounces associated with your Postmark server.

Parameters

- **api_key** – Your Postmark API key. Required, if *test* is not *True*.
- **secure** – Use the https scheme for the Postmark API. Defaults to *True*
- **test** – Use the Postmark Test API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *BounceTagsResponse*

`pystmark.get_delivery_stats` (*api_key=None, secure=None, test=None, **request_args*)

Get delivery stats for your Postmark account.

Parameters

- **api_key** – Your Postmark API key. Required, if *test* is not *True*.
- **secure** – Use the https scheme for the Postmark API. Defaults to *True*
- **test** – Use the Postmark Test API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *DeliveryStatsResponse*

4.2 Advanced API

4.2.1 Sending Email

class `pystmark.Sender` (*message=None, api_key=None, secure=True, test=False*)

Sends a single message via the Postmark API.

All of the arguments used in constructing this object are used as defaults in the final call to `Sender.send()`. You can override any of them at that time.

Parameters

- **message** (*dict* or *Message*) – Default message data, such as sender and reply_to.
- **api_key** – Your Postmark API key.
- **secure** – Use the https scheme for Postmark API. Defaults to *True*
- **test** – Make a test request to the Postmark API. Defaults to *False*.

response_class

alias of *SendResponse*

send (*message=None, api_key=None, secure=None, test=None, **request_args*)

Send request to Postmark API. Returns result of `requests.post()`.

Parameters

- **message** (*dict* or *Message*) – Your Postmark message data.
- **api_key** (*str*) – Your Postmark API key.
- **test** – Make a test request to the Postmark API.
- **secure** – Use the https Postmark API.

- **request_args** – Passed to `requests.post()`

Return type `requests.Response`

class `pystmark.BatchSender` (*message=None, api_key=None, secure=True, test=False*)

Sends a batch of messages via the Postmark API.

All of the arguments used in constructing this object are used as defaults in the final call to `BatchSender.send()`. You can override any of them at that time.

Parameters

- **message** (*dict* or *Message*) – Default message data, such as sender and reply_to.
- **api_key** – Your Postmark API key.
- **secure** – Use the https scheme for Postmark API. Defaults to *True*
- **test** – Make a test request to the Postmark API. Defaults to *False*.

response_class

alias of `BatchSendResponse`

send (*messages=None, api_key=None, secure=None, test=None, **request_args*)

Send batch request to Postmark API. Returns result of `requests.post()`.

Parameters

- **messages** (A list of *Message*) – Batch messages to send to the Postmark API.
- **api_key** – Your Postmark API key. Defaults to *self.api_key*.
- **test** – Make a test request to the Postmark API. Defaults to *self.test*.
- **secure** – Use the https Postmark API. Defaults to *self.secure*.
- **request_args** – Passed to `requests.request()`

Return type `BatchSendResponse`

class `pystmark.TemplateSender` (*message=None, api_key=None, secure=True, test=False*)

Sends a single message via the Postmark API with template.

All of the arguments used in constructing this object are used as defaults in the final call to `Sender.send()`. You can override any of them at that time.

Parameters

- **message** (*dict* or *Message*) – Default message data, such as sender and reply_to.
- **api_key** – Your Postmark API key.
- **secure** – Use the https scheme for Postmark API. Defaults to *True*
- **test** – Make a test request to the Postmark API. Defaults to *False*.

response_class

alias of `SendResponse`

send (*message=None, api_key=None, secure=None, test=None, **request_args*)

Send request to Postmark API. Returns result of `requests.post()`.

Parameters

- **message** (*dict* or *Message*) – Your Postmark message data.
- **api_key** (*str*) – Your Postmark API key.
- **test** – Make a test request to the Postmark API.

- **secure** – Use the https Postmark API.
- **request_args** – Passed to `requests.post()`

Return type `requests.Response`

4.2.2 Bounce Handling

class `pystmark.Bounces` (*api_key=None, secure=True, test=False*)
Multiple bounce retrieval endpoint wrapper.

Parameters

- **api_key** – Your Postmark API key. Defaults to *None*.
- **secure** – Use the https scheme for Postmark API. Defaults to *True*.
- **test** – Make a test request to the Postmark API. Defaults to *False*.

get (*bounce_type=None, inactive=None, email_filter=None, message_id=None, count=None, offset=None, api_key=None, secure=None, test=None, **request_args*)
Builds query string params from inputs. It handles offset and count defaults and validation.

Parameters

- **bounce_type** – The type of bounces retrieve. See *bounce_types* for a list of types, or read the Postmark API docs. Defaults to *None*.
- **inactive** – If *True*, retrieves inactive bounces only. Defaults to *None*.
- **email_filter** – A string to filter emails by. Defaults to *None*.
- **message_id** – Retrieve a bounce for a single message's ID. Defaults to *None*.
- **count** – The number of bounces to retrieve in this request. Defaults to 25 if *message_id* is not provided.
- **offset** – The page offset for bounces to retrieve. Defaults to 0 if *message_id* is not provided.
- **api_key** – Your Postmark API key. Defaults to *self.api_key*.
- **secure** – Use the https scheme for Postmark API. Defaults to *self.secure*.

Params test Use the Postmark test API. Defaults to *self.test*.

Return type `BouncesResponse`

response_class

alias of `BouncesResponse`

class `pystmark.Bounce` (*api_key=None, secure=True, test=False*)
Single bounce retrieval endpoint wrapper.

Parameters

- **api_key** – Your Postmark API key. Defaults to *None*.
- **secure** – Use the https scheme for Postmark API. Defaults to *True*.
- **test** – Make a test request to the Postmark API. Defaults to *False*.

get (*bounce_id, api_key=None, secure=None, test=None, **request_args*)
Retrieves a single bounce's data.

Parameters

- **bounce_id** – A bounce’s ID retrieved with *Bounces*.
- **api_key** – Your Postmark API key. Defaults to *self.api_key*.
- **secure** – Use the https scheme for Postmark API. Defaults to *self.secure*.
- **test** – Make a test request to the Postmark API. Defaults to *self.test*.
- **request_args** – Keyword args to pass to `requests.request()`.

Return type *BounceResponse*

response_class

alias of *BounceResponse*

class `pystmark.BounceDump` (*api_key=None, secure=True, test=False*)

Bounce dump endpoint wrapper.

get (*bounce_id, api_key=None, secure=None, test=None, **request_args*)

Retrieves a single bounce’s data.

Parameters

- **bounce_id** – A bounce’s ID retrieved with *Bounces*.
- **api_key** – Your Postmark API key. Defaults to *self.api_key*.
- **secure** – Use the https scheme for Postmark API. Defaults to *self.secure*.
- **test** – Make a test request to the Postmark API. Defaults to *self.test*.
- **request_args** – Keyword args to pass to `requests.request()`.

Return type *BounceResponse*

response_class

alias of *BounceDumpResponse*

class `pystmark.BounceActivate` (*api_key=None, secure=True, test=False*)

Bounce Activation endpoint wrapper.

Parameters

- **bounce_id** – A bounce’s ID retrieved with *Bounces*. Defaults to *None*.
- **api_key** – Your Postmark API key. Defaults to *None*.
- **secure** – Use the https scheme for Postmark API. Defaults to *True*.
- **test** – Make a test request to the Postmark API. Defaults to *False*.

activate (*bounce_id, api_key=None, secure=None, test=None, **request_args*)

Activates a bounce.

Parameters

- **bounce_id** – A bounce’s ID retrieved with *Bounces*.
- **api_key** – Your Postmark API key. Defaults to *self.api_key*.
- **secure** – Use the https scheme for Postmark API. Defaults to *self.secure*.
- **test** – Make a test request to the Postmark API. Defaults to *self.test*.
- **request_args** – Keyword args to pass to `requests.request()`.

Return type *BounceActivateResponse*

response_class
 alias of *BounceActivateResponse*

class `pystmark.BounceTags` (*api_key=None, secure=True, test=False*)
 Bounce tags endpoint wrapper.

get (*api_key=None, secure=None, test=None, **request_args*)
 Make a GET request to the Postmark API

Parameters

- **api_key** – Your Postmark API key.
- **secure** – Use the https scheme for Postmark API. Defaults to *True*
- **test** – Make a test request to the Postmark API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *Response*

response_class
 alias of *BounceTagsResponse*

class `pystmark.DeliveryStats` (*api_key=None, secure=True, test=False*)
 Delivery Stats endpoint wrapper.

get (*api_key=None, secure=None, test=None, **request_args*)
 Make a GET request to the Postmark API

Parameters

- **api_key** – Your Postmark API key.
- **secure** – Use the https scheme for Postmark API. Defaults to *True*
- **test** – Make a test request to the Postmark API. Defaults to *False*.
- **request_args** – Keyword arguments to pass to `requests.request()`.

Return type *Response*

response_class
 alias of *DeliveryStatsResponse*

4.2.3 Message Details

class `pystmark.OutboundMessageDetails` (*api_key=None, secure=True, test=False*)
 Outbound message details endpoint wrapper.

get (*message_id, api_key=None, secure=None, test=None, **request_args*)
 Retrieves a single messages's data.

Parameters

- **message_id** – A messages's ID.
- **api_key** – Your Postmark API key. Defaults to *self.api_key*.
- **secure** – Use the https scheme for Postmark API. Defaults to *self.secure*.
- **test** – Make a test request to the Postmark API. Defaults to *self.test*.
- **request_args** – Keyword args to pass to `requests.request()`.

Return type *OutboundMessageDetailsResponse*

response_class

alias of *OutboundMessageDetailsResponse*

4.3 Message Object

class `pystmark.Message` (*sender=None, to=None, cc=None, bcc=None, subject=None, template_id=None, template_alias=None, template_model=None, tag=None, html=None, text=None, reply_to=None, headers=None, attachments=None, verify=False, track_opens=None*)

A container for message(s) to send to the Postmark API. You can populate this message with defaults for initializing an *Interface*. The message will be combined with the final message and verified before transmission.

Parameters

- **sender** – Email address of the sender.
- **to** – Destination email address.
- **cc** – A list of cc'd email addresses.
- **bcc** – A list of bcc'd email address.
- **subject** – The message subject.
- **tag** – Tag your emails with this.
- **html** – HTML body content.
- **text** – Text body content.
- **reply_to** – Email address to reply to.
- **headers** (A list of *dict*, each with the keys 'Name' and 'Value'.) – Additional headers to include with the email. If you do not have the headers formatted for the Postmark API, use *Message.add_header()*.
- **attachments** (A list of *dict*, each with the keys 'Name', 'Content' and 'ContentType'.) – Attachments to include with the email. If you do not have the attachments formatted for the Postmark API, use *Message.attach_file()* or *Message.attach_binary()*.
- **verify** – Verify the message when initialized. Defaults to *False*.
- **track_opens** – Set to true to enable tracking email opens.

add_header (*name, value*)

Attach an email header to send with the message.

Parameters

- **name** – The name of the header value.
- **value** – The header value.

attach_binary (*data, filename, content_type=None, content_id=None*)

Attach a file to the message given raw binary data.

Parameters

- **data** – Raw data to attach to the message.
- **filename** – Name of the file for the data.
- **content_type** – mimetype of the data. It will be guessed from the filename if not provided.

- **content_id** – ContentID URL of the attachment. A RFC 2392- compliant URL for the attachment that allows it to be referenced from inside the body of the message. Must start with ‘cid:’

attach_file (*filename, content_type=None, content_id=None*)

Attach a file to the message given a filename.

Parameters

- **filename** – Name of the file to attach.
- **content_type** – mimetype of the data. It will be guessed from the filename if not provided.
- **content_id** – ContentID URL of the attachment. A RFC 2392- compliant URL for the attachment that allows it to be referenced from inside the body of the message. Must start with ‘cid:’

bcc

A comma delimited string of receivers for the message ‘Bcc’ field.

cc

A comma delimited string of receivers for the message ‘Cc’ field.

data ()

Returns data formatted for a POST request to the Postmark send API.

Return type *dict*

json ()

Return json-encoded string of message data.

Return type *str*

load_from (*other, **kwargs*)

Create a *Message* by merging *other* with *self*. Values from *other* will be copied to *self* if the value was not set on *self* and is set on *other*. :param other: The *Message* to copy defaults from. :type other: *Message*
:param kwargs: Additional keyword arguments to construct

Message with.

Return type *Message*

classmethod load_message (*message, **kwargs*)

Create a *Message* from a message data *dict*.

Parameters

- **message** – A *dict* of message data.
- **kwargs** – Additional keyword arguments to construct *Message* with.

Return type *Message*

recipients

A list of all recipients for this message.

to

A comma delimited string of receivers for the message ‘To’ field.

verify ()

Verifies the message data based on rules and restrictions defined in the Postmark API docs. There can be no more than 20 recipients in total. NOTE: This does not check that your attachments total less than 10MB, you must do that yourself.

4.4 Response Objects

class `pystmark.SendResponse` (*response*, *sender=None*)

Wrapper around `Sender.send()` and `BatchSender.send()`

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.
- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

class `pystmark.BatchSendResponse` (*response*, *sender=None*)

Wrapper around `Sender.send()` and `BatchSender.send()`

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.
- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

class `pystmark.BouncesResponse` (*response*, *sender=None*)

Wrapper for responses from `Bounces.get()`.

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.
- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

class `pystmark.BounceResponse` (*response*, *sender=None*)

Wrapper for responses from `Bounce.get()`.

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.
- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

class `pystmark.BounceDumpResponse` (*response*, *sender=None*)

Wrapper for responses from `BounceDump.get()`.

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.
- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

class `pystmark.BounceActivateResponse` (*response*, *sender=None*)

Wrapper for responses from the bounce activate endpoint.

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.

- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

class `pystmark.BounceTagsResponse` (*response*, *sender=None*)

Wrapper for responses from `BounceTags.get()`.

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.
- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

class `pystmark.DeliveryStatsResponse` (*response*, *sender=None*)

Wrapper for responses from `BounceActivate.activate()`.

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.
- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

class `pystmark.OutboundMessageDetailsResponse` (*response*, *sender=None*)

Wrapper for responses from `OutboundMessageDetails.get()`.

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.
- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

4.5 Response Data Wrappers

class `pystmark.MessageConfirmation` (*data*)

Wrapper around data returned from Postmark after sending

Parameters *data* – Data returned from Postmark upon sending a message

class `pystmark.BouncedMessage` (*bounce_data*, *sender=None*)

Bounced message data wrapper.

Parameters

- **bounce_data** – Raw bounced message data retrieved from `Bounce` or `Bounces`.
- **sender** – The *Interface* that made the request for the bounce data. Defaults to *None*.

dump (*sender=None*, ***kwargs*)

Retrieve raw email dump for this bounce.

Parameters

- **sender** – A `BounceDump` object to get dump with. Defaults to *None*.
- **kwargs** – Keyword arguments passed to `requests.request()`.

class `pystmark.BounceTypeData` (*bounce_type_data*)

Bounce type data wrapper.

Parameters `bounce_type_data` – Raw bounce type data retrieved from `DeliveryStats`.

4.6 Exceptions

class `pystmark.PystmarkError` (*message=None*)

Base *Exception* for `pystmark` errors.

Parameters `message` – Message to raise with the *Exception*. Defaults to *None*.

with_traceback ()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

class `pystmark.MessageError` (*message=None*)

Raised when a message meant to be sent to Postmark API looks malformed

with_traceback ()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

class `pystmark.BounceError` (*message=None*)

Raised when a bounce API method fails

with_traceback ()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

class `pystmark.ResponseError` (*response, message=None*)

Base *Exception* for errors received from Postmark API

Parameters

- `response` – A *Response*.
- `message` – Message to raise with the *Exception*. Defaults to *None*.

with_traceback ()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

class `pystmark.UnauthorizedError` (*response, message=None*)

Raised when Postmark responds with a `status_code` of 401 Indicates a missing or incorrect API key.

with_traceback ()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

class `pystmark.UnprocessableEntityError` (*response, message=None*)

Raised when Postmark responds with a `status_code` of 422. Indicates message(s) received by Postmark were malformed.

with_traceback ()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

class `pystmark.InternalServerError` (*response, message=None*)

Raised when Postmark responds with a `status_code` of 500 Indicates an error on Postmark's end. Any messages sent in the request were not received by them.

with_traceback ()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

4.7 Base Classes

class `pystmark.Interface` (*api_key=None, secure=True, test=False*)

Base class interface for Postmark API endpoint wrappers

Parameters

- **api_key** – Your Postmark API key. Defaults to *None*.
- **secure** – Use the https scheme for API requests. Defaults to *True*.
- **test** – Use the Postmark test API. Defaults to *False*.

response_class

alias of *Response*

class `pystmark.Response` (*response, sender=None*)

Base class for API response wrappers. The wrapped `requests.Response` object interface is exposed by this class, unless the attribute is defined in *self._attrs*.

Parameters

- **response** (`requests.Response`) – Response returned from `requests.request()`.
- **sender** (*Interface*) – The API interface wrapper that generated the request. Defaults to *None*.

raise_for_status()

Raise Postmark-specific HTTP errors. If there isn't one, the standard HTTP error is raised.

HTTP 401 raises *UnauthorizedError*

HTTP 422 raises *UnprocessableEntityError*

HTTP 500 raises *InternalServerError*

p

pystmark, 13

A

`activate()` (*pystmark.BounceActivate method*), 18
`activate_bounce()` (*in module pystmark*), 14
`add_header()` (*pystmark.Message method*), 20
`attach_binary()` (*pystmark.Message method*), 20
`attach_file()` (*pystmark.Message method*), 21

B

`BatchSender` (*class in pystmark*), 16
`BatchSendResponse` (*class in pystmark*), 22
`bcc` (*pystmark.Message attribute*), 21
`Bounce` (*class in pystmark*), 17
`BounceActivate` (*class in pystmark*), 18
`BounceActivateResponse` (*class in pystmark*), 22
`BouncedMessage` (*class in pystmark*), 23
`BounceDump` (*class in pystmark*), 18
`BounceDumpResponse` (*class in pystmark*), 22
`BounceError` (*class in pystmark*), 24
`BounceResponse` (*class in pystmark*), 22
`Bounces` (*class in pystmark*), 17
`BouncesResponse` (*class in pystmark*), 22
`BounceTags` (*class in pystmark*), 19
`BounceTagsResponse` (*class in pystmark*), 23
`BounceTypeData` (*class in pystmark*), 23

C

`cc` (*pystmark.Message attribute*), 21

D

`data()` (*pystmark.Message method*), 21
`DeliveryStats` (*class in pystmark*), 19
`DeliveryStatsResponse` (*class in pystmark*), 23
`dump()` (*pystmark.BouncedMessage method*), 23

G

`get()` (*pystmark.Bounce method*), 17
`get()` (*pystmark.BounceDump method*), 18
`get()` (*pystmark.Bounces method*), 17
`get()` (*pystmark.BounceTags method*), 19

`get()` (*pystmark.DeliveryStats method*), 19
`get()` (*pystmark.OutboundMessageDetails method*), 19
`get_bounce()` (*in module pystmark*), 14
`get_bounce_dump()` (*in module pystmark*), 14
`get_bounce_tags()` (*in module pystmark*), 14
`get_bounces()` (*in module pystmark*), 14
`get_delivery_stats()` (*in module pystmark*), 15

I

`Interface` (*class in pystmark*), 25
`InternalServerError` (*class in pystmark*), 24

J

`json()` (*pystmark.Message method*), 21

L

`load_from()` (*pystmark.Message method*), 21
`load_message()` (*pystmark.Message class method*), 21

M

`Message` (*class in pystmark*), 20
`MessageConfirmation` (*class in pystmark*), 23
`MessageError` (*class in pystmark*), 24

O

`OutboundMessageDetails` (*class in pystmark*), 19
`OutboundMessageDetailsResponse` (*class in pystmark*), 23

P

`pystmark` (*module*), 1, 7, 13
`PystmarkError` (*class in pystmark*), 24

R

`raise_for_status()` (*pystmark.Response method*), 25
`recipients` (*pystmark.Message attribute*), 21
`Response` (*class in pystmark*), 25

`response_class` (*pystmark.BatchSender* attribute), 16
`response_class` (*pystmark.Bounce* attribute), 18
`response_class` (*pystmark.BounceActivate* attribute), 18
`response_class` (*pystmark.BounceDump* attribute), 18
`response_class` (*pystmark.Bounces* attribute), 17
`response_class` (*pystmark.BounceTags* attribute), 19
`response_class` (*pystmark.DeliveryStats* attribute), 19
`response_class` (*pystmark.Interface* attribute), 25
`response_class` (*pystmark.OutboundMessageDetails* attribute), 19
`response_class` (*pystmark.Sender* attribute), 15
`response_class` (*pystmark.TemplateSender* attribute), 16
`ResponseError` (class in *pystmark*), 24

S

`send()` (in module *pystmark*), 13
`send()` (*pystmark.BatchSender* method), 16
`send()` (*pystmark.Sender* method), 15
`send()` (*pystmark.TemplateSender* method), 16
`send_batch()` (in module *pystmark*), 13
`Sender` (class in *pystmark*), 15
`SendResponse` (class in *pystmark*), 22

T

`TemplateSender` (class in *pystmark*), 16
`to` (*pystmark.Message* attribute), 21

U

`UnauthorizedError` (class in *pystmark*), 24
`UnprocessableEntityError` (class in *pystmark*), 24

V

`verify()` (*pystmark.Message* method), 21

W

`with_traceback()` (*pystmark.BounceError* method), 24
`with_traceback()` (*pystmark.InternalServerError* method), 24
`with_traceback()` (*pystmark.MessageError* method), 24
`with_traceback()` (*pystmark.PystmarkError* method), 24
`with_traceback()` (*pystmark.ResponseError* method), 24