

---

# mailgun Documentation

*Release 0.1.4*

**AI2**

**Jul 19, 2019**



---

## Contents

---

<b>1</b>	<b>Install</b>	<b>1</b>
<b>2</b>	<b>Examples</b>	<b>3</b>
<b>3</b>	<b>Examples</b>	<b>5</b>
3.1	Install . . . . .	5
3.2	Mailgun Examples . . . . .	5
<b>4</b>	<b>Mailgun Class</b>	<b>7</b>
4.1	Mailgun Class . . . . .	7
<b>5</b>	<b>Mailgun Utils</b>	<b>11</b>
5.1	Mailgun Utils . . . . .	11
<b>6</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



# CHAPTER 1

---

## Install

---

```
pip install mailgun3_python
```



## CHAPTER 2

---

### Examples

---

```
from mailgun import Mailgun

mailgun = Mailgun()

# send email with attachment
mailgun.send_message("Text, <test@gmail.com>",
                    "bkrm.dahal@gmail.com",
                    "Docsumo: Automate invoice data capture at Docsumo",
                    'Hi, welcome to docsumo.',
                    files=['./invoice.pdf'])

# send message with saved template
mailgun.send_message_template("Text, <test@gmail.com>",
                              "bkrm.dahal@gmail.com",
                              "Docsumo: Automate invoice data capture at Docsumo",
                              "welcome_email",
                              {"company_name": "docsumo", "first_name": "bikram"})
```





### 3.1 Install

```
pip install mailgun3_python
```

### 3.2 Mailgun Examples

```
from mailgun import Mailgun

mailgun = Mailgun()

# send email with attachment
mailgun.send_message("Text, <test@gmail.com>",
                    "bkrm.dahal@gmail.com",
                    "Docsumo: Automate invoice data capture at Docsumo",
                    'Hi, welcome to docsumo.',
                    files=['./invoice.pdf'])

# send message with saved template
mailgun.send_message_template("Text, <test@gmail.com>",
                              "bkrm.dahal@gmail.com",
                              "Docsumo: Automate invoice data capture at Docsumo",
                              "welcome_email",
                              {"company_name": "docsumo", "first_name": "bikram"})
```



### 4.1 Mailgun Class

#### 4.1.1 Mailgun

Mailgun class to send email and get extracted data

```
class mailgun.mailgun.Mailgun (apikey: str = None, domain: str = None, version: str = 'v3')  
    Bases: object
```

Initializes an object of Mailgun class.

**Args:**

**apiKey: str** API key provided from mailgun. You can also save env var as MAILGUN\_API\_KEY .

**domain: str** domain to use for this class. You can also save env var as MAILGUN\_DOMAIN .

**version: str** Mailgun API version.

**Returns:** Mailgun class object.

```
get_logs (params: dict = {'event': 'stored'})  
    Logs
```

**Args:**

**Params: dict** filter for logs.

**Return:** logs dicts: dict

```
get_message_mime (url: str)  
    get email mime to parse email
```

**Args:**

**url: str** image storage url

**Return:** json with body-mime: dict

**mailing\_list\_add\_email** (*list\_name: str, data: dict*)

Add user to mailing list

**Args:** list\_name: str data: dict

data of user

```
{'subscribed': True,
 'address': 'bar@example.com',
 'name': 'Bob Bar',
 'description': 'Developer',
 'vars': '{"age": 26}'}
```

**Return:** Responses from API: dict

**mailing\_list\_create** (*list\_name: str, description: str*)

Create new mailing list

**Args:** list\_name: str description: str

**Return:** Response dict: dict

**mailing\_list\_delete** (*list\_name: str*)

Delete mailing list

**Args:** list\_name: str

**Return:** Response dict: dict

**mailing\_list\_delete\_email** (*list\_name: str, email: str*)

Delete user from mailing list

**Args:** list\_name: str email: str

email to be removed

**Return:** Responses from API: dict

**mailing\_list\_update\_email** (*list\_name: str, email: str, data: dict*)

Update user to detail on mailing list

**Args:** list\_name: str email: str data: dict

data of user

```
{'subscribed': True,
 'name': 'Bob Bar',
 'description': 'Developer',
 'vars': '{"age": 26}'}
```

**Return:** Responses from API: dict

**parse\_email\_mime** (*body\_mime, email\_id: str = None, save\_attachment\_dir: str = 'tmp'*)

parse email-mime

**Args:** body-mime: str

**Return:** Metadata and file saved in tmp dir: dict

**send\_message** (*sender\_email: str, to: str, subject: str, html\_body: str = None, text\_body: str = None, files: list = None, extra\_data: dict = None*)

Send email

**Args:**

**sender\_email:** str example “Test email <test@docsumo.com>”

**to:** `str` user email id

**subject:** `str` **html\_body:** `str` **text\_body:** `str` **files:** `list`  
list of files

**extra\_data:** `dict` extra data for tagging and tracking

**Return:** Response from API: `json`

```
{
  'id': '<20190714191959.1.D7B53FD6D5077D03@mg.docsumo.com>',
  'message': 'Queued. Thank you.'
}
```

Example:

```
send_message("Text, <test@gmail.com>",
             "bkrm.dahal@gmail.com",
             "Docsumo: Automate invoice data capture at Docsumo",
             "welcome_email",
             files=["./data.pdf"])
```

**send\_message\_template** (*sender\_email: str, to: str, subject: str, template\_name: str, data: dict,*  
*extra\_data: dict = None*)

Send email using template

**Args:**

**sender\_email:** `str` example "Test email <test@docsumo.com>"

**to:** `str` user email id

**subject:** `str` **template\_name:** `str` **data:** `str`  
data for template make string from dict using `json.dumps`

**extra\_data:** `dict` extra data for tagging and tracking

**Return:** Response from API: `json`

```
{
  'id': '<20190714191959.1.D7B53FD6D5077D03@mg.docsumo.com>',
  'message': 'Queued. Thank you.'
}
```

Example:

```
send_message_template("Text, <test@gmail.com>",
                      "bkrm.dahal@gmail.com",
                      "Docsumo: Automate invoice data capture at Docsumo",
                      "welcome_email",
                      {"company_name": "docsumo", "first_name": "bikram"})
```

**validated\_email** (*email: str*)

Validate the Email

**Args:** `email: str`

**Return:** valid True or false: `bool`



### 5.1 Mailgun Utils

Parse body-mime

`mailgun.email_parsing.parse_email(email_string: str, email_id: str, output_dir: str = 'tmp')`  
parse email and save the file

**Args:**

**email\_string: str** string of email body-mime

**email\_id: str** unique id for email

**output\_dir: str** folder for attachments

**Return:** Email metadata dict: dict

```
{
  "from": bkrm.dahal@gmail.com,
  "to": bikram.dahal@docsumo.com,
  "date": "2019-01-01 22:00:00",
  "subject": "Testing email"
}
```

**exception** `mailgun.error.NoAPIKey`

Bases: Exception

**exception** `mailgun.error.NoDomain`

Bases: Exception





## CHAPTER 6

---

### Indices and tables

---

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



### m

`mailgun.email_parsing`, [11](#)  
`mailgun.error`, [11](#)  
`mailgun.mailgun`, [7](#)



### G

`get_logs()` (*mailgun.mailgun.Mailgun method*), 7  
`get_message_mime()` (*mailgun.mailgun.Mailgun method*), 7

### M

`Mailgun` (*class in mailgun.mailgun*), 7  
`mailgun.email_parsing` (*module*), 11  
`mailgun.error` (*module*), 11  
`mailgun.mailgun` (*module*), 7  
`mailing_list_add_email()` (*mailgun.mailgun.Mailgun method*), 7  
`mailing_list_create()` (*mailgun.mailgun.Mailgun method*), 8  
`mailing_list_delete()` (*mailgun.mailgun.Mailgun method*), 8  
`mailing_list_delete_email()` (*mailgun.mailgun.Mailgun method*), 8  
`mailing_list_update_email()` (*mailgun.mailgun.Mailgun method*), 8

### N

`NoAPIKey`, 11  
`NoDomain`, 11

### P

`parse_email()` (*in module mailgun.email\_parsing*), 11  
`parse_email_mime()` (*mailgun.mailgun.Mailgun method*), 8

### S

`send_message()` (*mailgun.mailgun.Mailgun method*), 8  
`send_message_template()` (*mailgun.mailgun.Mailgun method*), 9

### V

`validated_email()` (*mailgun.mailgun.Mailgun method*), 9