

```

1 # Source: Oppia - https://github.com/oppia/oppia - an online learning tool
  that enables anyone to easily create and share interactive activities
2
3 """Provides mailgun api to send email."""
4
5 from __future__ import absolute_import # pylint: disable=import-only-modules
6 from __future__ import unicode_literals # pylint: disable=import-only-
  modules
7
8 import base64
9
10 from core.platform.email import gae_email_services
11 import feconf
12 import python_utils
13
14
15 def post_to_mailgun(data):
16     """Send POST HTTP request to mailgun api. This method is adopted from
17     the requests library's post method.
18     Args:
19         - data: dict. The data to be sent in the request's body.
20     Returns:
21         Response from the server. The object is a file-like object.
22         https://docs.python.org/2/library/urllib2.html
23     """
24     if not feconf.MAILGUN_API_KEY:
25         raise Exception('Mailgun API key is not available.')
26
27     if not feconf.MAILGUN_DOMAIN_NAME:
28         raise Exception('Mailgun domain name is not set.')
29
30     encoded = base64.b64encode(b'api:%s' % feconf.MAILGUN_API_KEY).strip()
31     auth_str = 'Basic %s' % encoded
32     header = {'Authorization': auth_str}
33     server = (
34         'https://api.mailgun.net/v3/%s/messages' %
35         feconf.MAILGUN_DOMAIN_NAME)
36     data = python_utils.url_encode(data)
37     req = python_utils.url_request(server, data, header)
38     return python_utils.url_open(req)
39
40 def send_mail(
41     sender_email, recipient_email, subject, plaintext_body,
42     html_body, bcc_admin=False, reply_to_id=None):
43     """Sends an email using mailgun api.
44     In general this function should only be called from
45     email_manager._send_email().
46     Args:
47         - sender_email: str. the email address of the sender. This should be in
48           the form 'SENDER_NAME <SENDER_EMAIL_ADDRESS>'.
49         - recipient_email: str. the email address of the recipient.
50         - subject: str. The subject line of the email.
51         - plaintext_body: str. The plaintext body of the email.
52         - html_body: str. The HTML body of the email. Must fit in a datastore
53           entity.
54         - bcc_admin: bool. Whether to bcc feconf.ADMIN_EMAIL_ADDRESS on the
55           email.
56         - reply_to_id: str. The unique id of the sender.
57     Raises:

```

```

57     Exception: if the configuration in feconf.py forbids emails from being
58     sent.
59     Exception: if mailgun api key is not stored in feconf.MAILGUN_API_KEY.
60     Exception: if mailgun domain name is not stored in
61     feconf.MAILGUN_DOMAIN_NAME.
62     (and possibly other exceptions, due to mail.send_mail() failures)
63     """
64     if not feconf.CAN_SEND_EMAILS:
65         raise Exception('This app cannot send emails to users.')
66
67     data = {
68         'from': sender_email,
69         'to': recipient_email,
70         'subject': subject.encode(encoding='utf-8'),
71         'text': plaintext_body.encode(encoding='utf-8'),
72         'html': html_body.encode(encoding='utf-8')
73     }
74
75     if bcc_admin:
76         data['bcc'] = feconf.ADMIN_EMAIL_ADDRESS
77
78     if reply_to_id:
79         reply_to = gae_email_services.get_incoming_email_address(reply_to_id)
80         data['h:Reply-To'] = reply_to
81
82     post_to_mailgun(data)
83
84
85 def send_bulk_mail(
86     sender_email, recipient_emails, subject, plaintext_body, html_body):
87     """Sends an email using mailgun api.
88     In general this function should only be called from
89     email_manager._send_email().
90     Args:
91         - sender_email: str. the email address of the sender. This should be in
92           the form 'SENDER_NAME <SENDER_EMAIL_ADDRESS>'.
93         - recipient_emails: list. list of the email addresses of recipients.
94         - subject: str. The subject line of the email.
95         - plaintext_body: str. The plaintext body of the email.
96         - html_body: str. The HTML body of the email. Must fit in a datastore
97           entity.
98         - reply_to_id: str or None. The unique reply-to id used in reply-to
99           email
100           sent to recipient.
101     Raises:
102         Exception: if the configuration in feconf.py forbids emails from being
103         sent.
104         Exception: if mailgun api key is not stored in feconf.MAILGUN_API_KEY.
105         Exception: if mailgun domain name is not stored in
106         feconf.MAILGUN_DOMAIN_NAME.
107         (and possibly other exceptions, due to mail.send_mail() failures)
108     """
109     if not feconf.CAN_SEND_EMAILS:
110         raise Exception('This app cannot send emails to users.')
111
112     # To send bulk emails we pass list of recipients in 'to' paarameter of
113     # post data. Maximum limit of recipients per request is 1000.
114     # For more detail check following link:
115     # https://documentation.mailgun.com/user_manual.html#batch-sending
116     recipient_email_sets = [

```

```
116     recipient_emails[i:i + 1000]
117     for i in python_utils.RANGE(0, len(recipient_emails), 1000)]
118
119     for email_set in recipient_email_sets:
120         # 'recipient-variable' in post data forces mailgun to send individual
121         # email to each recipient (This is intended to be a workaround for
122         # sending individual emails).
123         data = {
124             'from': sender_email,
125             'to': email_set,
126             'subject': subject.encode(encoding='utf-8'),
127             'text': plaintext_body.encode(encoding='utf-8'),
128             'html': html_body.encode(encoding='utf-8'),
129             'recipient-variables': '{}'}
130
131     post_to_mailgun(data)
132
```