SENDGRID INTEGRATION WITH PYTHON

|  |  |
| --- | --- |
| Date | 14 Nov 2022 |
| Team ID | PNT2022TMID19691 |
| Project Name | NUTRITION ASSISTANT APPLICATION |

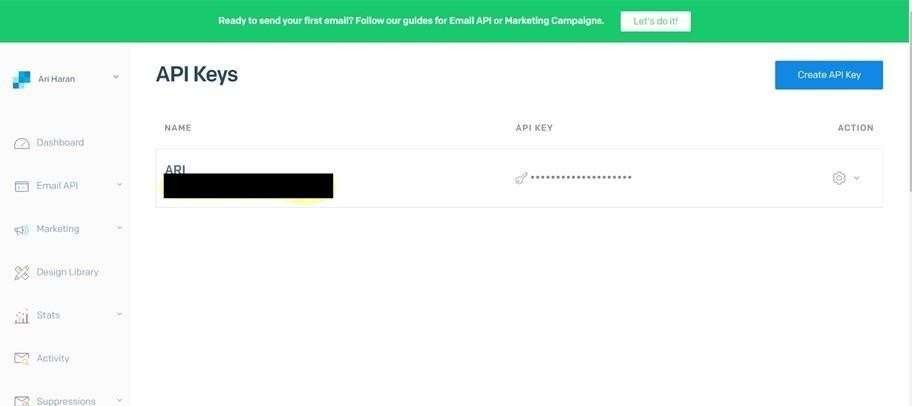
**STEP 1:**

Requirements:

**Python 2.6, 2.7, 3.4 or 3.5.**

**STEP 2:**

Creating an API key



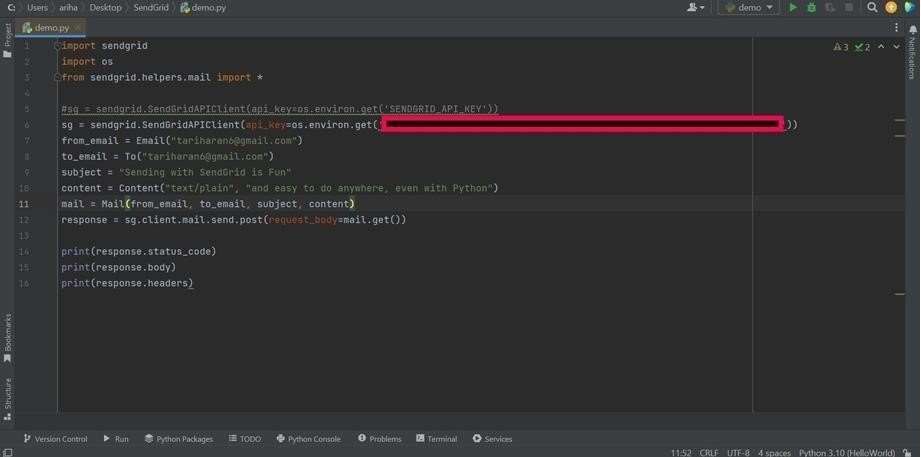
# STEP 3:

INSTALL

# PAKAGE: > pip install sendgrid

SETP 4:

# SEND EMAIL



**SENDGRID PYTHON CODE :**

1

2

3

4

5

6

7

8

9

10

"""HTTP Client library""" import json

import logging

from .exceptions import handle\_error

try:

# Python 3

import urllib.request as urllib

from urllib.parse import urlencode from urllib.error import HTTPError

11 except ImportError:

12

# Python 2

1. **import os**
2. **from sendgrid import SendGridAPIClient**
3. **from sendgrid.helpers.mail import Mail 4**

**5**

**6**

**7**

**8**

**9**

**10**

**11**

**12**

**13**

**14**

**17**

**message = Mail( from\_email='from\_email@example.com', to\_emails='to@example.com',**

**subject='Sending with Twilio SendGrid is Fun', html\_content='<strong>and easy to do anywhere, even with Python</strong>')**

**try:**

**sg = SendGridAPIClient(os.environ.get('SENDGRID\_API\_KEY')) response = sg.send(message)**

**print(response.status\_code)**

**print(response.body) 15 print(response.headers) 16 except Exception as e:**

**print(e.message)**

**HTTP CLIENT PROGRAM:**

import urllib2 as urllib

14

15

16

from urllib2 import HTTPError

from urllib import urlencode

17 \_logger = logging.getLogger( name ) 18

19

20

21

23

24

25

26

27

28

29

30

31

32

33

class Response(object):

"""Holds the response from an API call.""" 22

def init (self, response):

"""

:param response: The return value from a open call

on a urllib.build\_opener()

:type response: urllib response object """

self.\_status\_code = response.getcode() self.\_body = response.read()

self.\_headers = response.info()

@property

|  |
| --- |
| 34 def status\_code(self):  35 """  36 :return: integer, status code of API call 37 """  38 return self.\_status\_code 39   1. @property 2. def body(self):   42 """  43 :return: response from the API  44 """  45 return self.\_body  46  47 @property |

@property

def to\_dict(self):

"""

:return: dict of response from the API """

if self.body:

return json.loads(self.body.decode('utf-8')) else:

return None

class Client(object):

"""Quickly and easily access any REST or REST-like API.""" 67 # These are the supported HTTP verbs

def headers(self):

"""

:return: dict of response headers """

return self.\_headers

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

68

69

71

72

73

74

75

76

78

79

methods = {'delete', 'get', 'patch', 'post', 'put'} 70 def init (self,

host, request\_headers=None, version=None,

url\_path=None,

append\_slash=False, 77

"""

timeout=None):

80

81

:param host: Base URL for the api. (e.g. https://api.sendgrid.com)

:type host: string

:param request\_headers: A dictionary of the headers you want

applied on all calls

83

84

85

:type request\_headers: dictionary

:param version: The version number of the API.

Subclass \_build\_versioned\_url for custom

behavior.

86

87

88

89

90

91

92

93

94

95

96

Or just pass the version as part of the URL (e.g. client.\_("/v3"))

:type version: integer

:param url\_path: A list of the url path segments

:type url\_path: list of strings """

self.host = host

self.request\_headers = request\_headers or {} self.\_version = version

# \_url\_path keeps track of the dynamically built url

self.\_url\_path = url\_path or []

def \_build\_versioned\_url(self, url):

# APPEND SLASH set

self.append\_slash = append\_slash self.timeout = timeout

97

98

99

100

101

102

"""Subclass this function for your own needs.

113

114

:param query\_params: A dictionary of all the query

|  |  |  |
| --- | --- | --- |
| 103 | Or just pass the version as part of | the URL |
| 104 | (e.g. client.\_('/v3')) |  |
| 105 | :param url: URI portion of the full | URL being requested |
| 106 | :type url: string |  |
| 107 | :return: string |  |
| 108 | """ |  |
| 109  110 | return '{}/v{}{}'.format(self.host, url) | str(self.\_version), |
| 111 | def \_build\_url(self, query\_params): |  |
| 112 | """Build the final URL to be passed to | urllib |

parameters

|  |  |  |
| --- | --- | --- |
| 115 | :type query\_params: dictionary |  |
| 116 | :return: string |
| 117 | """ |
| 118 | url = '' |
| 119 | count = 0 |
| 120 | while count < len(self.\_url\_path): |
| 121 | url += '/{}'.format(self.\_url\_path[count]) |
| 122 | count += 1 |
| 123 |  |
| 124 | # add slash |
| 125 | if self.append\_slash: |
| 126 | url += '/' |
| 127 |  |
| 128 | if query\_params: |
| 129 | url\_values = urlencode(sorted(query\_params.items()), | True) |
| 130 | url = '{}?{}'.format(url, url\_values) |  |
| 131 |  |  |
| 132 | if self.\_version: |  |
| 133 | url = self.\_build\_versioned\_url(url) |  |
| 134 | else: |  |
| 135 | url = '{}{}'.format(self.host, url) |  |
| 136 | return url |  |
| 137 |  |  |
| 138 | def \_update\_headers(self, request\_headers): |  |
| 139 | """Update the headers for the request |  |
| 140 |  |  |
| 141 | :param request\_headers: headers to set for the API call |  |

1. :type request\_headers: dictionary
2. :return: dictionary

144 """

145 self.request\_headers.update(request\_headers) 146

147 def \_build\_client(self, name=None):

|  |  |  |
| --- | --- | --- |
| 148 | """Make a new Client object | |
| 149 |  | |
| 150 | :param name: Name of the url segment | |
| 151 | :type name: string | |
| 152 | :return: A Client object | |
| 153 | """ | |
| 154 | url\_path = self.\_url\_path + [name] if name else | |
| self.\_url\_path  155 return Client(host=self.host,   1. version=self.\_version, 2. request\_headers=self.request\_headers, 3. url\_path=url\_path, 4. append\_slash=self.append\_slash, 5. timeout=self.timeout) 161 6. def \_make\_request(self, opener, request,   timeout=None):   1. """Make the API call and return the response. is   This separated into testing. | | |
| 164 | it's own function, so we | can mock it easily for |
| 165 |  |  |
| 166 | :param opener: |  |

|  |
| --- |
| 1. :type opener: 2. :param request: url payload to request 3. :type request: urllib.Request object 4. :param timeout: timeout value or None 5. :type timeout: float 6. :return: urllib response   173 """   1. timeout = timeout or self.timeout 2. try: 3. return opener.open(request, timeout=timeout) 4. except HTTPError as err: 5. exc = handle\_error(err) 6. exc. cause = None 7. \_logger.debug('{method} Response: {status} |

:return: string, version

"""

214

215

216

217

218

self.\_version = args[0] return self.\_build\_client()

return get\_version

219

220

221

222

223

224

225

226

227

228

229

# We have reached the end of the method chain, make the API call

if name in self.methods:

method = name.upper()

230

231

def http\_request( request\_body=None, query\_params=None, request\_headers=None, timeout=None,

\*\*\_):

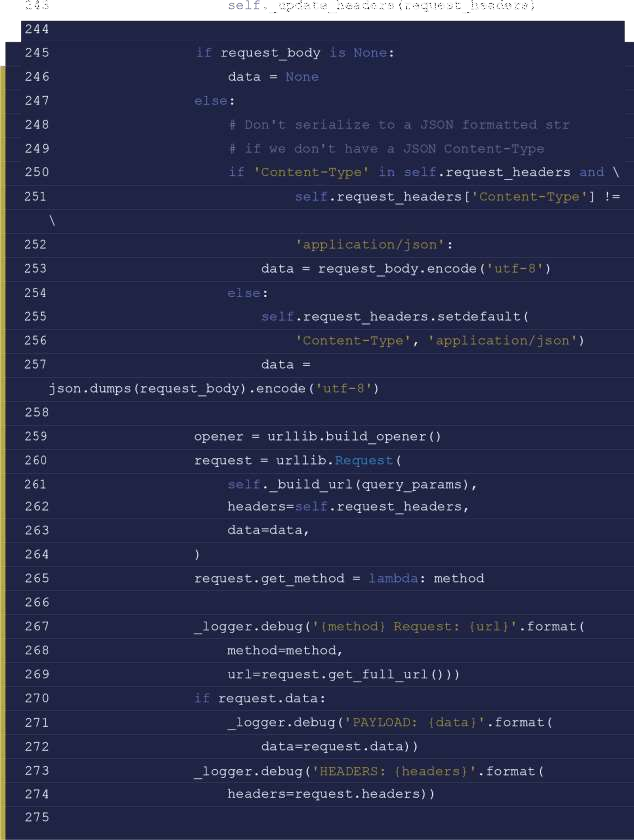
"""Make the API call

:param timeout: HTTP request timeout. Will be propagated to

urllib client

:type timeout: float

|  |  |  |
| --- | --- | --- |
| 232 |  | :param request\_headers: HTTP headers. Will be |
| merged | into |  |
| 233 |  | current client object state |
| 234 |  | :type request\_headers: dict |
| 235 |  | :param query\_params: HTTP query parameters |
| 236 |  | :type query\_params: dict |
| 237 |  | :param request\_body: HTTP request body |
| 238 |  | :type request\_body: string or json-serializable |
| object | | |
| 239 | :param kwargs: | |
| 240 | :return: Response object | |
| 241 | """ | |
| 242 | if request\_headers: | |



response = Response(

self.\_make\_request(opener, request,

timeout=timeout)

278 )

279

280 \_logger.debug('{method} Response: {status}

{body}'.format(

295

def setstate (self, state):

|  |  |  |
| --- | --- | --- |
| 281 |  | method=method, |
| 282 |  | status=response.status\_code, |
| 283 |  | body=response.body)) |
| 284 |  |  |
| 285 |  | return response |
| 286 |  |  |
| 287 | return | http\_request 288 |
|  | else: |  |
| 289 |  | # Add a segment to the URL |
| 290 |  | return self.\_(name) |
| 291 |  |  |
| 292 |  | def getstate (self): |
| 293 |  | return self. dict |
| 294 |  |  |