**SENDGRID INTEGRATION WITH PYTHON**

|  |  |
| --- | --- |
| Date | 14 Nov 2022 |
| Team ID | PNT2022TMID26524 |
| Project Name | NUTRITION ASSISTANT APPLICATION |

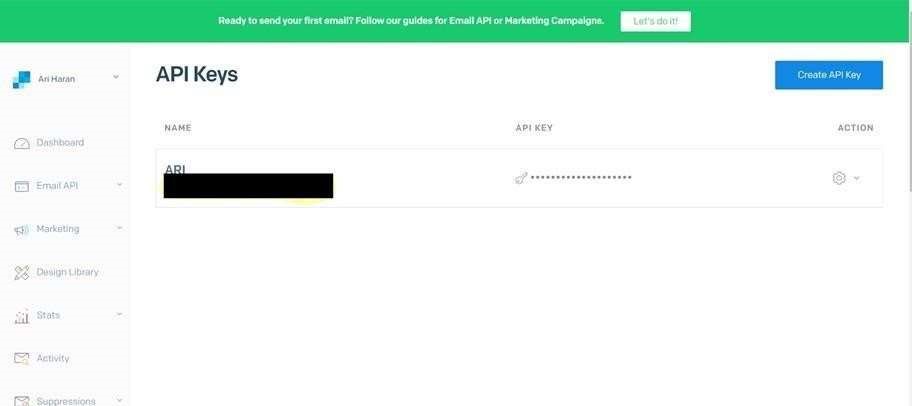
**STEP 1:**

Requirements:

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

**STEP 2:**

Creating an API key

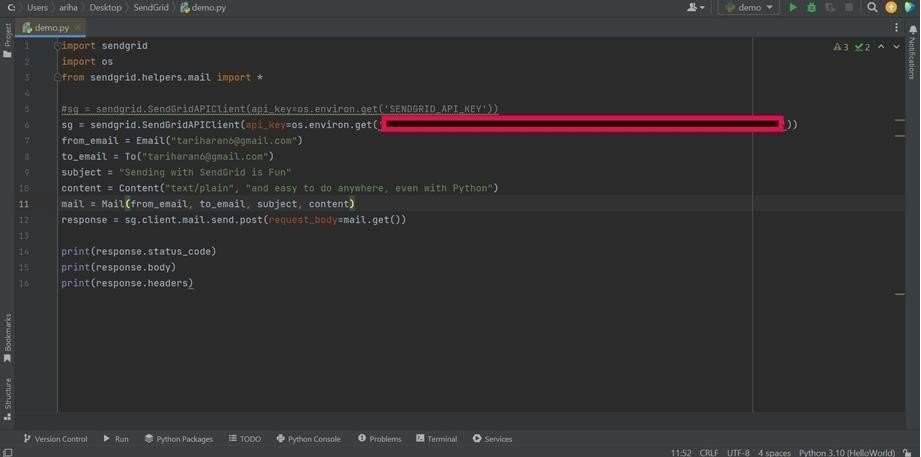


STEP 3: INSTALL

PAKAGE: > pip install sendgrid

SETP 4:

SEND EMAIL



**SENDGRID PYTHON CODE :**

|  |  |  |
| --- | --- | --- |
| 1. """HTTP Client library""" 2. import json 3. import logging 4. from .exceptions import handle\_error   5   1. try: 2. # Python 3 3. import urllib.request as urllib 4. from urllib.parse import urlencode 5. from urllib.error import HTTPError 6. except ImportError: 7. # Python 2 | | |
| **1** | | **import os** |
| **2** | | **from sendgrid import SendGridAPIClient** |
| **3** | | **from sendgrid.helpers.mail import Mail** |
| **4**  **5** | | **message = Mail(** |
| **6** | | **from\_email='from\_email@example.com',** |
| **7** | | **to\_emails='to@example.com',** |
| **8** | | **subject='Sending with Twilio SendGrid is Fun',** |
| **9** | | **html\_content='<strong>and easy to do anywhere, even with**  **Python</strong>')** |
| **10** | | **try:** |
| **11** | | **sg = SendGridAPIClient(os.environ.get('SENDGRID\_API\_KEY'))** |
| **12** | | **response = sg.send(message)** |
| **13** | | **print(response.status\_code)** |
| **14** | | **print(response.body) 15 print(response.headers) 16 except Exception as e:** |
| **17** | | **print(e.message)** |

**HTTP CLIENT PROGRAM:**

13 import urllib2 as urllib

|  |
| --- |
| 1. from urllib2 import HTTPError 2. from urllib import urlencode   16  17 \_logger = logging.getLogger( name )  18  19   1. class Response(object): 2. """Holds the response from an API call.""" 22 3. def init (self, response): 4. """ 5. :param response: The return value from a   open call   1. on a urllib.build\_opener() 2. :type response: urllib response object 3. """ 4. self.\_status\_code = response.getcode() 5. self.\_body = response.read() 6. self.\_headers = response.info()   32  33 @property |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 34 | def status\_code(self): | | 35 | """ | | 36 | :return: integer, status code of API call | | 37 | """ | | 38 | return self.\_status\_code | | 39  40 | @property | | 41 | def body(self): | | 42 | """ | | 43 | :return: response from the API | | 44 | """ | | 45 | return self.\_body | | 46 47 | @property | |

|  |  |
| --- | --- |
|  |  |
| 48 | def headers(self): |
| 49 | """ |
| 50 | :return: dict of response headers |
| 51 | """ |
| 52 | return self.\_headers |
| 53  54 | @property |
| 55 | def to\_dict(self): |
| 56 | """ |
| 57 | :return: dict of response from the API |
| 58 | """ |
| 59 | if self.body: |
| 60 | return json.loads(self.body.decode('utf-8')) |
| 61 | else: |
| 62 | return None |
| 63  64  65 | class Client(object): |
| 66 | """Quickly and easily access any REST or REST-like API.""" 67 |
| 68 | # These are the supported HTTP verbs |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | 69 methods = {'delete', 'get', 'patch', 'post', 'put'} 70   1. def init (self, 2. host, 3. request\_headers=None, 4. version=None, 5. url\_path=None, 6. append\_slash=False, 77 timeout=None): 7. """ 8. :param host: Base URL for the api. (e.g.   https://api.sendgrid.com)   1. :type host: string 2. :param request\_headers: A dictionary of the headers you want |   82 applied on all calls   |  |  | | --- | --- | | 83 | :type request\_headers: dictionary | | 84 | :param version: The version number of the API. | | 85 | Subclass \_build\_versioned\_url for custom | | behavior.  86 | Or just pass the version as part of the URL | | 87 | (e.g. client.\_("/v3")) | | 88 | :type version: integer | | 89 | :param url\_path: A list of the url path segments | | 90 | :type url\_path: list of strings | | 91 | """ | | 92 | self.host = host | | 93 | self.request\_headers = request\_headers or {} | | 94 | self.\_version = version | | 95 | # \_url\_path keeps track of the dynamically built url | | 96 | self.\_url\_path = url\_path or [] | |

|  |  |
| --- | --- |
|  |  |
| 97 | # APPEND SLASH set |
| 98 | self.append\_slash = append\_slash |
| 99 | self.timeout = timeout |
| 100 101 | def \_build\_versioned\_url(self, url): |

|  |  |  |
| --- | --- | --- |
| 102 |  | """Subclass this function for your own needs. |
| 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 |  | return '{}/v{}{}'.format(self.host, str(self.\_version), url) |
| 110  111 |  | def \_build\_url(self, query\_params): |
| 112 |  | """Build the final URL to be passed to urllib |
| 113 114 |  | :param query\_params: A dictionary of all the query |

|  |  |
| --- | --- |
| pa | rameters |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  | |  |
|  | | | |  |
| 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 |
|  | |  | | |
| 142 | | :type request\_headers: dictionary | | |
| 143 | | :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,

156

version=

self

.\_version,

157

request\_headers=

self

.request\_headers,

158

url\_path=url\_path,

159

append\_slash=

self

.append\_slash,

160

timeout=

self

.timeout)

161

162

def

\_make\_request(

self

, opener, request,

timeout=

None

):

163

"""Make the API call and return the response.

This separated into

164

it

's own function, so we can mock it easily for

165

166

:

param opener

:

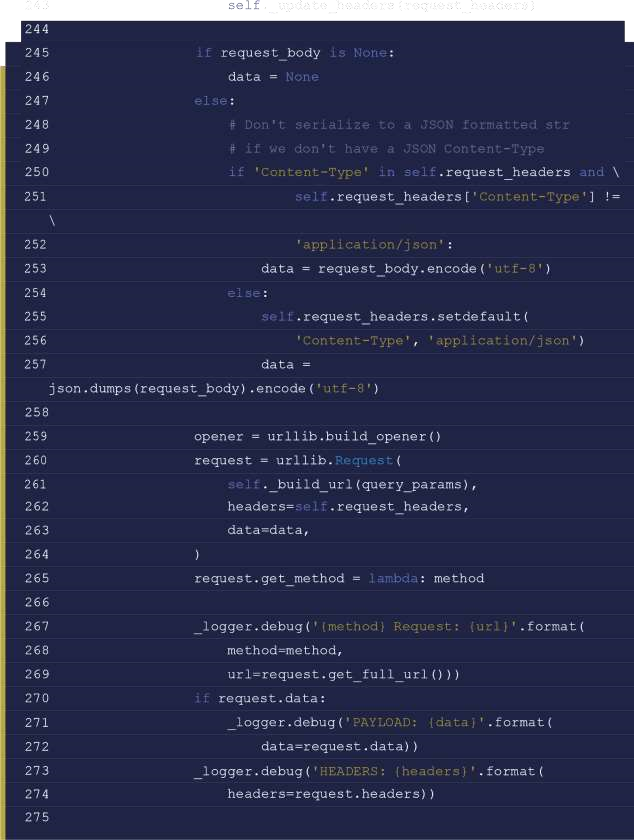
is

testing.

|  |  |
| --- | --- |
| 167 | :type opener: |
| 168 | :param request: url payload to request |
| 169 | :type request: urllib.Request object |
| 170 | :param timeout: timeout value or None |
| 171 | :type timeout: float |
| 172 | :return: urllib response |
| 173 | """ |
| 174 | timeout = timeout or self.timeout |
| 175 | try: |
| 176 | return opener.open(request, timeout=timeout) |
| 177 | except HTTPError as err: |
| 178 | exc = handle\_error(err) |
| 179 | exc. cause = None |
| 180 | \_logger.debug('{method} Response: {status} |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 212    :  return  :    string  , version      213 """   |  |  |  | | --- | --- | --- | | 214 |  | self.\_version = args[0] | | 215 |  | return self.\_build\_client() | | 216 |  | return get\_version | | 217  218 |  | # We have reached the end of the method chain, make the  API call | | 219 |  | if name in self.methods: | | 220 |  | method = name.upper() | | 221  222 |  | def http\_request( | | 223 |  | request\_body=None, | | 224 |  | query\_params=None, | | 225 |  | request\_headers=None, | | 226 |  | timeout=None, | | 227 |  | \*\*\_): | | 228 |  | """Make the API call | | 229 |  | :param timeout: HTTP request timeout. Will be propagated to | | 230 |  | urllib client | | 231 |  | :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: | |



276

response =

Response

(

277

self

.\_make\_request(opener, request,

timeout=timeout)

278

)

279

280

\_logger.debug(

'{method} Response: {status}

{

body}'

.format(

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

295

def

setstate (

self

, state):