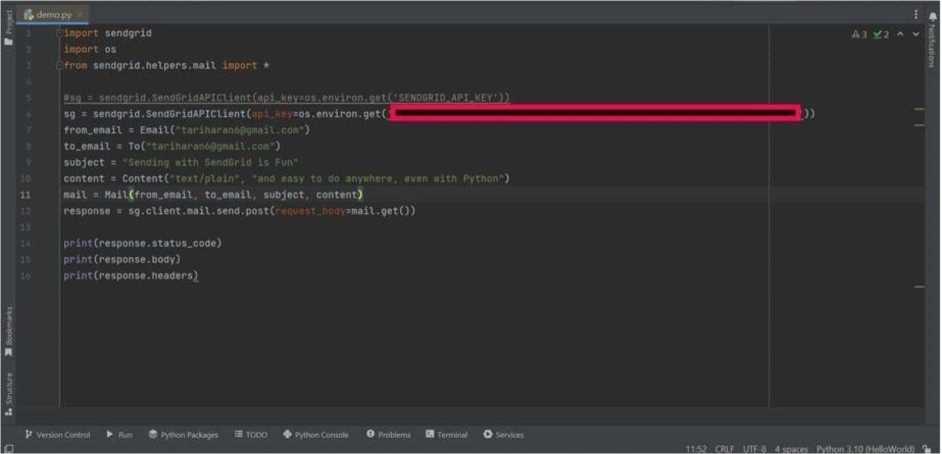
INTEGRATING SEND GRID WITH PYTHON FLASK

|  |  |
| --- | --- |
| Date | 10 NOV 2022 |
| Team ID | PNT2022TMID50454 |
| Project Name | Nutrition Assistant Application |

Creating API key:





SENDGRID PYTHON CODE :

|  |  |  |
| --- | --- | --- |
| 1 | **xmport os** |  |
| 2 | **from sendgrid 3 endGridAPIC \_** |  |
| 3 | **sendgrid. helper a .mail :<ni.l** |  |
| 4 |  |  |
| 5 | **message = >’： (** |  |
| 6 | **from emaxl= from email^example.com'** | ***T*** |
| 7 | **to\_ema±ls= to@example . com\* t** |  |
| 8 | **subject='Sending with Twilio Send** | **Grid 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 | **.(response.status^code)** |  |
| 14 | **print(response.body)** |  |
| 15 | **(response.headers)** |  |
| 16 | **except Exception as e ：** |  |
| 17 | **r?r in• (e .message)** |  |

HTTP CLIENT PROGRAM

1. ir:c：rt json
2. lr.c : rc logging
3. .exceptions handle\_error

5

1. try:
2. . \ ■ urllib.request urllib
3. :• urllib.parse - urlencode
4. i . ' urllib.error - - ■ L ： ：■
5. .:
6. Pynhon .

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

1. , - urllib2 .. urllib
2. : urllib2 imp c LTTPEiror
3. . urllib l:-; :rt urlencode

16

1. \_logger = logging. getLogger ( name\_\_)

18

19

20 ( ):

1. "”"Holds the response from an API call."”"

22

1. . init ( ： t response):
2. "••••
3. :param response: : - value ：. a open call
4. on a urllib.build\_opener()
5. :type response: urllib response

2 g •>

1. r .\_\_status\_\_code » response. getcode ()
2. s:-:・\_body = response.read()
3. .—headers = response.info()

32

^property

status—code( ):

:i ..: integer, status code of API call "\*1 "

.\_status\_code

^property

body( ):

"""

:- response the API

\*( h *n*

: e 1 r .\_body

@property

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

( ):

"""Quickly and easily access any REST or REST-like API.'””'

67

68

69

♦ These are the supported HTTP verbs

methods = (\* delete1 r , \* patci . , , ' I

70

71

72

host.

73

74

request—headers= ,

version: ,

75

76

77

url\_path= ,

append\_\_slash= ,

timeout: ):

78

H II \*1

headers ( ):

n n

:- . :diet of response headers

H,• W

.. t：. - • .—headers

^property

s -1 to\_dict( ：):

w «• n

:： ..:diet of response the API

H «• tt

.--f.body:

json.loads( :•body.decode

return None

:param host: URL the api. (e.g.

79

https:

1. :type host: ■ :
2. :param request\_headers: A dictionary of the headers you want

|  |  |
| --- | --- |
| 82 | applied on all calls |
| 83 | :type request headers: dictionary |
| 84 | :param version: version number of the API. |
| 85 | build versioned url for custom |
| behavior• | |
| 86 | just the version 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 | w •• •» |
| 92 | .host = host |
| 93 | :.request—headers = request—headers (} |
| 94 | :• If・ version = version |
| 95 | # url path keeps track of the dynamically built url |
| 96 | :♦\_url\_path = url—path -[] |
| 97 | 1 APPEND SLASH Bet |
| 98 | • -..append slash = append slash |
| 99 | .timeout = timeout |
| 100 |  |
| 101 | build versioned url( , url): |
| 102 | MWMSubclass tins function for your own needs. |
| 103 | just the version part of the URL |
| 104 | (e.g. client.-('/v3’)) |
| 105 | :param url: URI portion of the full URL being requested |
| 106 | :type url: |
| 107 | :\_eturn: .. j |
| 108 | •\* N M |
| 109 | '.format( .host, str< i^ version), |
| url) |  |
| 110 |  |
| 111 | . —build—url( , query—params): |
| 112 | ""••Build the final URL to be passed to urllib |
| 113 |  |
| 114 | :param query—params: A dictionary of all the query |

parameters

1. :type query\_params: dictionary
2. : ： - turn: irng

117

1. url ="
2. count = 0
3. count < len (： •• .\_url\_path):
4. url += ’ ’ .format ( .\_url\_path[count])
5. count += 1

123

1. ***i*** Add D
2. .append\_slash:
3. url +=・/•

127

1. - query\_params:
2. url\_values = urlencode(sorted(query\_params.items())r

True)

1. url =・ ’ .format (urlr url\_values)

131

1. xf self•—version:
2. url = 1J <\_build\_versioned\_url(url)
3. els :
4. url = \*0(1’ .format (self .host, url)
5. return url

137

1. \_update\_headers( , request\_headers):
2. •• ""Update the headers for the request

140

1. :param request\_headers: headers to the API call
2. :type request—headers: dictionary
3. :return: dictionary

***Y44 •»«»•***

1. .:. request^headers .update (request\_headers)

146

1. build client( f name= ):

|  |  |  |
| --- | --- | --- |
| 148 | ’•’’"Make a new Client object 1 | |
| 149 |  |  |
| 150 | :param name: 1:๘ of the url segment |  |
| 151 | :type name: .. |  |
| 152 | ：I ： A |  |
| 153 | n w |  |
| 154 | url\_path = \*\_url\_path + [name] . name |  |
| self. | \_url\_path |  |
| 155 | 1-..■. ui：. 1 ieuL (host= .host, |  |
| 156 | version=se.:♦—version. |  |
| 157 | request headers: .request headers. |  |
| 158 | url\_\_path=url\_path, |  |
| 159 | append slash= .:.append slash. |  |
| 160 | timeout: .timeout) |  |
| 161 |  |  |
| 162 | make request( , opener, request, timeout= >: |  |
| 163 | ••"••Make the API call and return the response. This | is |
| I separated into | |  |
| 164 | it \* ***s*** own function, so we can mock it. easily for | testing. |
| 165 |  |  |
| 166 | :param opener: |  |
| 167 | :type opener: |  |
| 168 | :param request: url payload to request |  |
| 169 | :type request: urllib. |  |
| 170 | :param timeout: timeout value . |  |
| 171 | :type timeout: |  |
| 172 | :- :urllib response |  |
| 173 | •tn" |  |
| 174 | timeout = timeout - l . - If.timeout |  |
| 175 | try: |  |
| 176 | opener.open(request, timeout=timeout) |  |
| 177 | err: |  |
| 178 | exc = handle\_error(err) |  |
| 179 | exc. cause = |  |
| 180 | —logger .debug」 ?.•\_ s[\_ \_r.se : |  |

|  |  |
| --- | --- |
| (body)’ | ♦ format( |
| 181 | method=request.get\_\_method(), |
| 182 | status=exc^status\_\_coder |
| 183 | body=exc.body)) |
| 184 | ***i：ii-*** exc |
| 185 |  |
| 186 . f ( -Ifr name): | |
| 187 | ,,H'\*Add variable values to the url. |
| 188 | (e.g. /your/api/(variable\_value}/call) |
| 189 | example: you have a reserved word. |
| such a: |  |
| 190 | your urlr you must method. |
| 191 |  |
| 192 | :param name: .. of the url segment |
| 193 | :type name: |
| 194 | '1 . • |
| 195 |  |
| 196 | re’ ..-\_1 .\_build\_client (name) |
| 197 |  |
| 198 - getattr ( ..r name): | |
| 199 | •’"•’Dynamically add method calls to the url, then call a |
| method. |  |
| 200 | <e.g・ client.name.name.method()) |
| 201 | can also add a version number ..nr.c |
| <version(<：： >) | |
| 202 |  |
| 203 | :param name: of the url segment method call |
| 204 | :type name: - integer name == version |
| 205 | :• -: mixed |
| 206 | n n «• |
| 207 | .:name = ・version’: |
| 208 | get\_version(\*argsr \*\*kwargs): |
| 209 | «• tt M |
| 210 | :param args: diet of settings |
| 211 | :param kwargs: unused |

|  |  |  |  |
| --- | --- | --- | --- |
| 212 |  | :. .： f version |  |
| 213 |  | •• •• M |  |
| 214 |  | :. version = args[] |  |
| 215 |  | -i.\_build\_client() |  |
| 216 | get\_version | |  |
| 217 |  |  |  |
| 218 | # We have reached the end of the method chaxn, make the | | |
|  |  |  |  |
| 219 | i f name | .methods: |  |
| 220 | method = name.upper() | |  |
| 221 |  |  |  |
| 222 | def | http\_request( |  |
| 223 |  | r eques t\_body=: i c ：, |  |
| 224 |  | que r y\_pa r ams = r |  |
| 225 |  | request headers: r |  |
| 226 |  | timeout:!: , |  |
| 227 |  |  |  |
| 228 |  | •'•’"Make the API call |  |
| 229 |  | :param timeout: HTTP request timeout. | be |
| propagated to | |  |  |
| 230 |  | urllib client |  |
| 231 |  | :type timeout: |  |
| 232 |  | :param request headers: HTTP headers. | be |
| merged | into |  |  |
| 233 |  | current client state |  |
| 234 |  | :type request\_headers: diet |  |
| 235 |  | :param query params: HTTP query parameters |  |
| 236 |  | :type query\_params: diet |  |
| 237 |  | :param request—body: HTTP request body |  |
| 238 |  | :type request\_body: ! json-seriali2able | |
| object |  |  |  |
| 239 |  | :param kwargs: |  |
| 240 |  | :return: Response |  |
| 241 |  | N " •• |  |
| 242 |  | request headers: |  |

API

244

245

246

247

248

249

250

251

252

253

254

255

256

257

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

i: requestJbody :: :

data = Nor.­

:

* Don't serxalixe to a JSON formatted str
* If we don・t have a JSON Content-Type

' ’ /l- • .request—headers \

:.request\_headers [' Ccnt<f：.t-Type \* ] !=

’ : -on ’:

data = request\_body.encode<•uLf-J'»

•i i::、:

\*request—headers.setdefaultj

\*Content-Type•, 1application/j son9) data =

json.dumps(request\_body).encode(‘ i> ’)

258

opener = urllib.build—opener() request = urllib.： (

-\_.—build—url(query\_params), headers: .request—headers,

data=datar

)

request.get—method = : method

\_logger.debug (’ : j .• -• . ： ’ . format (

method=methodr

url=request.get—full—url())) request.data:

-logger. debug < ‘ ； ;'iL\_ a: ： • . format (

data=request.data))

\_logger.debug (’ HEADERS ： (heac：L：i s '. format ( headers=request.headers))

1. response = Response(
2. sell <\_make\_\_request(opener, request,

timeout=timeout)

1. )

279

280

logger.debug('imethod) Response: {status)

i . format (

method=method.

281

1. status=response.status\_code,
2. body=response.body))

284

1. re Lurn response

286

1. return http\_request
2. else:
3. -.
4. r ■\_□!! - f .\_(name)

291

1. . getstate ( ：):
2. - . -. . diet

294

295 setstate ( , state):