import os

import json

from sendgrid import SendGridAPIClient

from sendgrid.helpers.mail import \*

# NOTE: you will need move this file to the root

# directory of this project to execute properly.

def build\_hello\_email():

## Send a Single Email to a Single Recipient

message = Mail(from\_email=From('from@example.com', 'Example From Name'),

to\_emails=To('to@example.com', 'Example To Name'),

subject=Subject('Sending with SendGrid is Fun'),

plain\_text\_content=PlainTextContent('and easy to do anywhere, even with Python'),

html\_content=HtmlContent('<strong>and easy to do anywhere, even with Python</strong>'))

try:

print(json.dumps(message.get(), sort\_keys=True, indent=4))

return message.get()

except SendGridException as e:

print(e.message)

mock\_personalization = Personalization()

personalization\_dict = get\_mock\_personalization\_dict()

for cc\_addr in personalization\_dict['cc\_list']:

mock\_personalization.add\_to(cc\_addr)

for bcc\_addr in personalization\_dict['bcc\_list']:

mock\_personalization.add\_bcc(bcc\_addr)

for header in personalization\_dict['headers']:

mock\_personalization.add\_header(header)

for substitution in personalization\_dict['substitutions']:

mock\_personalization.add\_substitution(substitution)

for arg in personalization\_dict['custom\_args']:

mock\_personalization.add\_custom\_arg(arg)

mock\_personalization.subject = personalization\_dict['subject']

mock\_personalization.send\_at = personalization\_dict['send\_at']

message.add\_personalization(mock\_personalization)

return message

def get\_mock\_personalization\_dict():

"""Get a dict of personalization mock."""

mock\_pers = dict()

mock\_pers['to\_list'] = [To("test1@example.com",

"Example User"),

To("test2@example.com",

"Example User")]

mock\_pers['cc\_list'] = [To("test3@example.com",

"Example User"),

To("test4@example.com",

"Example User")]

mock\_pers['bcc\_list'] = [To("test5@example.com"),

To("test6@example.com")]

mock\_pers['subject'] = ("Hello World from the Personalized "

"SendGrid Python Library")

mock\_pers['headers'] = [Header("X-Test", "test"),

Header("X-Mock", "true")]

mock\_pers['substitutions'] = [Substitution("%name%", "Example User"),

Substitution("%city%", "Denver")]

mock\_pers['custom\_args'] = [CustomArg("user\_id", "343"),

CustomArg("type", "marketing")]

mock\_pers['send\_at'] = 1443636843

return mock\_pers

def build\_multiple\_emails\_personalized():

# Note that the domain for all From email addresses must match

message = Mail(from\_email=From('from@example.com', 'Example From Name'),

subject=Subject('Sending with SendGrid is Fun'),

plain\_text\_content=PlainTextContent('and easy to do anywhere, even with Python'),

html\_content=HtmlContent('<strong>and easy to do anywhere, even with Python</strong>'))

mock\_personalization = Personalization()

mock\_personalization.add\_to(To('test@example.com', 'Example User 1'))

mock\_personalization.add\_cc(Cc('test1@example.com', 'Example User 2'))

message.add\_personalization(mock\_personalization)

mock\_personalization\_2 = Personalization()

mock\_personalization\_2.add\_to(To('test2@example.com', 'Example User 3'))

mock\_personalization\_2.set\_from(From('from@example.com', 'Example From Name 2'))

mock\_personalization\_2.add\_bcc(Bcc('test3@example.com', 'Example User 4'))

message.add\_personalization(mock\_personalization\_2)

try:

print(json.dumps(message.get(), sort\_keys=True, indent=4))

return message.get()

except SendGridException as e:

print(e.message)

return message

def build\_attachment1():

"""Build attachment mock. Make sure your content is base64 encoded before passing into attachment.content.

Another example: https://github.com/sendgrid/sendgrid-python/blob/HEAD/use\_cases/attachment.md"""

attachment = Attachment()

attachment.file\_content = ("TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQsIGNvbnNl"

"Y3RldHVyIGFkaXBpc2NpbmcgZWxpdC4gQ3JhcyBwdW12")

attachment.file\_type = "application/pdf"

attachment.file\_name = "balance\_001.pdf"

attachment.disposition = "attachment"

attachment.content\_id = "Balance Sheet"

return attachment

def build\_attachment2():

"""Build attachment mock."""

attachment = Attachment()

attachment.file\_content = "BwdW"

attachment.file\_type = "image/png"

attachment.file\_name = "banner.png"

attachment.disposition = "inline"

attachment.content\_id = "Banner"

return attachment

def build\_kitchen\_sink():

"""All settings set"""

from sendgrid.helpers.mail import (

Mail, From, To, Cc, Bcc, Subject, PlainTextContent,

HtmlContent, SendGridException, Substitution,

Header, CustomArg, SendAt, Content, MimeType, Attachment,

FileName, FileContent, FileType, Disposition, ContentId,

TemplateId, Section, ReplyTo, Category, BatchId, Asm,

GroupId, GroupsToDisplay, IpPoolName, MailSettings,

BccSettings, BccSettingsEmail, BypassListManagement,

FooterSettings, FooterText, FooterHtml, SandBoxMode,

SpamCheck, SpamThreshold, SpamUrl, TrackingSettings,

ClickTracking, SubscriptionTracking, SubscriptionText,

SubscriptionHtml, SubscriptionSubstitutionTag,

OpenTracking, OpenTrackingSubstitutionTag, Ganalytics,

UtmSource, UtmMedium, UtmTerm, UtmContent, UtmCampaign)

import time

import datetime

message = Mail()

# Define Personalizations

message.to = To('test1@sendgrid.com', 'Example User1', p=0)

message.to = [

To('test2@sendgrid.com', 'Example User2', p=0),

To('test3@sendgrid.com', 'Example User3', p=0)

]

message.cc = Cc('test4@example.com', 'Example User4', p=0)

message.cc = [

Cc('test5@example.com', 'Example User5', p=0),

Cc('test6@example.com', 'Example User6', p=0)

]

message.bcc = Bcc('test7@example.com', 'Example User7', p=0)

message.bcc = [

Bcc('test8@example.com', 'Example User8', p=0),

Bcc('test9@example.com', 'Example User9', p=0)

]

message.subject = Subject('Sending with SendGrid is Fun 0', p=0)

message.header = Header('X-Test1', 'Test1', p=0)

message.header = Header('X-Test2', 'Test2', p=0)

message.header = [

Header('X-Test3', 'Test3', p=0),

Header('X-Test4', 'Test4', p=0)

]

message.substitution = Substitution('%name1%', 'Example Name 1', p=0)

message.substitution = Substitution('%city1%', 'Example City 1', p=0)

message.substitution = [

Substitution('%name2%', 'Example Name 2', p=0),

Substitution('%city2%', 'Example City 2', p=0)

]

message.custom\_arg = CustomArg('marketing1', 'true', p=0)

message.custom\_arg = CustomArg('transactional1', 'false', p=0)

message.custom\_arg = [

CustomArg('marketing2', 'false', p=0),

CustomArg('transactional2', 'true', p=0)

]

message.send\_at = SendAt(1461775051, p=0)

message.to = To('test10@example.com', 'Example User10', p=1)

message.to = [

To('test11@example.com', 'Example User11', p=1),

To('test12@example.com', 'Example User12', p=1)

]

message.cc = Cc('test13@example.com', 'Example User13', p=1)

message.cc = [

Cc('test14@example.com', 'Example User14', p=1),

Cc('test15@example.com', 'Example User15', p=1)

]

message.bcc = Bcc('test16@example.com', 'Example User16', p=1)

message.bcc = [

Bcc('test17@example.com', 'Example User17', p=1),

Bcc('test18@example.com', 'Example User18', p=1)

]

message.header = Header('X-Test5', 'Test5', p=1)

message.header = Header('X-Test6', 'Test6', p=1)

message.header = [

Header('X-Test7', 'Test7', p=1),

Header('X-Test8', 'Test8', p=1)

]

message.substitution = Substitution('%name3%', 'Example Name 3', p=1)

message.substitution = Substitution('%city3%', 'Example City 3', p=1)

message.substitution = [

Substitution('%name4%', 'Example Name 4', p=1),

Substitution('%city4%', 'Example City 4', p=1)

]

message.custom\_arg = CustomArg('marketing3', 'true', p=1)

message.custom\_arg = CustomArg('transactional3', 'false', p=1)

message.custom\_arg = [

CustomArg('marketing4', 'false', p=1),

CustomArg('transactional4', 'true', p=1)

]

message.send\_at = SendAt(1461775052, p=1)

message.subject = Subject('Sending with SendGrid is Fun 1', p=1)

# The values below this comment are global to entire message

message.from\_email = From('help@twilio.com', 'Twilio SendGrid')

message.reply\_to = ReplyTo('help\_reply@twilio.com', 'Twilio SendGrid Reply')

message.subject = Subject('Sending with SendGrid is Fun 2')

message.content = Content(MimeType.text, 'and easy to do anywhere, even with Python')

message.content = Content(MimeType.html, '<strong>and easy to do anywhere, even with Python</strong>')

message.content = [

Content('text/calendar', 'Party Time!!'),

Content('text/custom', 'Party Time 2!!')

]

message.attachment = Attachment(FileContent('base64 encoded content 1'),

FileName('balance\_001.pdf'),

FileType('application/pdf'),

Disposition('attachment'),

ContentId('Content ID 1'))

message.attachment = [

Attachment(FileContent('base64 encoded content 2'),

FileName('banner.png'),

FileType('image/png'),

Disposition('inline'),

ContentId('Content ID 2')),

Attachment(FileContent('base64 encoded content 3'),

FileName('banner2.png'),

FileType('image/png'),

Disposition('inline'),

ContentId('Content ID 3'))

]

message.template\_id = TemplateId('13b8f94f-bcae-4ec6-b752-70d6cb59f932')

message.section = Section('%section1%', 'Substitution for Section 1 Tag')

message.section = [

Section('%section2%', 'Substitution for Section 2 Tag'),

Section('%section3%', 'Substitution for Section 3 Tag')

]

message.header = Header('X-Test9', 'Test9')

message.header = Header('X-Test10', 'Test10')

message.header = [

Header('X-Test11', 'Test11'),

Header('X-Test12', 'Test12')

]

message.category = Category('Category 1')

message.category = Category('Category 2')

message.category = [

Category('Category 1'),

Category('Category 2')

]

message.custom\_arg = CustomArg('marketing5', 'false')

message.custom\_arg = CustomArg('transactional5', 'true')

message.custom\_arg = [

CustomArg('marketing6', 'true'),

CustomArg('transactional6', 'false')

]

message.send\_at = SendAt(1461775053)

message.batch\_id = BatchId("HkJ5yLYULb7Rj8GKSx7u025ouWVlMgAi")

message.asm = Asm(GroupId(1), GroupsToDisplay([1,2,3,4]))

message.ip\_pool\_name = IpPoolName("IP Pool Name")

mail\_settings = MailSettings()

mail\_settings.bcc\_settings = BccSettings(False, BccSettingsTo("bcc@twilio.com"))

mail\_settings.bypass\_list\_management = BypassListManagement(False)

mail\_settings.footer\_settings = FooterSettings(True, FooterText("w00t"), FooterHtml("<string>w00t!<strong>"))

mail\_settings.sandbox\_mode = SandBoxMode(True)

mail\_settings.spam\_check = SpamCheck(True, SpamThreshold(5), SpamUrl("https://example.com"))

message.mail\_settings = mail\_settings

tracking\_settings = TrackingSettings()

tracking\_settings.click\_tracking = ClickTracking(True, False)

tracking\_settings.open\_tracking = OpenTracking(True, OpenTrackingSubstitutionTag("open\_tracking"))

tracking\_settings.subscription\_tracking = SubscriptionTracking(

True,

SubscriptionText("Goodbye"),

SubscriptionHtml("<strong>Goodbye!</strong>"),

SubscriptionSubstitutionTag("unsubscribe"))

tracking\_settings.ganalytics = Ganalytics(

True,

UtmSource("utm\_source"),

UtmMedium("utm\_medium"),

UtmTerm("utm\_term"),

UtmContent("utm\_content"),

UtmCampaign("utm\_campaign"))

message.tracking\_settings = tracking\_settings

return message

def send\_multiple\_emails\_personalized():

# Assumes you set your environment variable:

# https://github.com/sendgrid/sendgrid-python/blob/HEAD/TROUBLESHOOTING.md#environment-variables-and-your-sendgrid-api-key

message = build\_multiple\_emails\_personalized()

sendgrid\_client = SendGridAPIClient(os.environ.get('SENDGRID\_API\_KEY'))

response = sendgrid\_client.send(message=message)

print(response.status\_code)

print(response.body)

print(response.headers)

def send\_hello\_email():

# Assumes you set your environment variable:

# https://github.com/sendgrid/sendgrid-python/blob/HEAD/TROUBLESHOOTING.md#environment-variables-and-your-sendgrid-api-key

message = build\_hello\_email()

sendgrid\_client = SendGridAPIClient(os.environ.get('SENDGRID\_API\_KEY'))

response = sendgrid\_client.send(message=message)

print(response.status\_code)

print(response.body)

print(response.headers)

def send\_kitchen\_sink():

# Assumes you set your environment variable:

# https://github.com/sendgrid/sendgrid-python/blob/HEAD/TROUBLESHOOTING.md#environment-variables-and-your-sendgrid-api-key

message = build\_kitchen\_sink()

sendgrid\_client = SendGridAPIClient(os.environ.get('SENDGRID\_API\_KEY'))

response = sendgrid\_client.send(message=message)

print(response.status\_code)

print(response.body)

print(response.headers)

## this will actually send an email

# send\_hello\_email()

## this will send multiple emails

# send\_multiple\_emails\_personalized()

## this will only send an email if you set SandBox Mode to False

# send\_kitchen\_sink()