Search projects

Help Sponsor Log in Register

microsoftgraph-python 0.1.7

✓ Latest version

Released: Mar 10, 2020

```
pip install microsoftgraph-python
```

API wrapper for Microsoft Graph written in Python

Navigation

Project description

Release history

Download files

Project links

Homepage

Project description

microsoft-python

Microsoft graph API wrapper for Microsoft Graph written in Python.

Installing

```
pip install microsoftgraph-python
```

Usage

If you need an office 365 token, send office365 attribute in True like this:

Statistics

GitHub statistics:

- Stars: 24
- Forks: 17
- Open issues/PRs: 3

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery

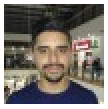
Meta

License: MIT

Author: Miguel Ferrer, Nerio Rincon, Yordy Gelvez

Maintainers

gearplug


ingmferrer

```
from microsoftgraph.client import Client
client = Client('CLIENT_ID', 'CLIENT_SECRET', account_t
```

If you don't, just instance the library like this:

```
from microsoftgraph.client import Client
client = Client('CLIENT_ID', 'CLIENT_SECRET', account_t
```

Get authorization url

```
url = client.authorization_url(redirect_uri, scope, sta
```

Exchange the code for an access token

```
token = client.exchange_code(redirect_uri, code)
```

Refresh token

```
token = client.refresh_token(redirect_uri, refresh_toke
```

Set token

```
token = client.set_token(token)
```

Get me

```
me = client.get_me()
```

Get message

```
me = client.get_message(message_id="")
```

Webhook section, see the api documentation: https://developer.microsoft.com/en-us/graph/docs/api-reference/beta/resources/webhooks

Create subscription

```
subscription = client.create_subscription(change_type,
```

Renew subscription

```
renew = client.renew_subscription(subscription_id, expi
```

Delete subscription

```
renew = client.delete_subscription(subscription_id)
```

Onenote section, see the api documentation: https://developer.microsoft.com/en-us/graph/docs/concepts/integrate_with_onenote

List notebooks

```
notebooks = client.list_notebooks()
```

Get notebook

```
notebook = client.get_notebook(notebook_id)
```

Get notebook sections

```
section_notebook = client.get_notebook_sections(noteboo
```

Create page

```
add_page = client.create_page(section_id, files)
```

List pages

```
pages = client.list_pages()
```

Calendar section, see the api documentation: https://developer.microsoft.com/en-us/graph/docs/api-reference/beta/resources/calendar

Get events

```
events = client.get_me_events()
```

Create calendar event

```
events = client.create_calendar_event(subject, content,
                              recurrence_type, recurren
                              recurrence_range_startdat
```

Get calendars

```
events = client.get_me_calendars()
```

Create calendar

```
events = client.create_calendar(name)
```

Contacts section, see the api documentation: https://developer.microsoft.com/en-us/graph/docs/api-reference/beta/resources/contact

Get contacts

If you need a specific contact send the contact id in data_id

```
specific_contact = client.outlook_get_me_contacts(data_
```

If you want all the contacts

```
specific_contact = client.outlook_get_me_contacts()
```

Create contact

```
add_contact = client.outlook_create_me_contact()
```

Create contact in specific folder

```
add_contact_folder = client.outlook_create_contact_in_f
```

Get contact folders

```
folders = client.outlook_get_contact_folders()
```

Create contact folders

```
add_folders = client.outlook_create_contact_folder()
```

Onedrive section, see the api documentation: https://developer.microsoft.com/en-us/graph/docs/api-reference/beta/resources/onedrive

Get root items

```
root_items = client.drive_root_items()
```

Get root children items

```
root_children_items = client.drive_root_children_items(
```

Get specific folder items

```
folder_items = client.drive_specific_folder(folder_id)
```

Excel section, see the api documentation: https://developer.microsoft.com/en-us/graph/docs/api-reference/beta/resources/excel

For use excel, you should know the folder id where the file is

Create session for specific item

```
create_session = client.drive_create_session(item_id)
```

Refresh session for specific item

```
refresh_session = client.drive_refresh_session(item_id)
```

Close session for specific item

```
close_session = client.drive_close_session(item_id)
```

Download the contents of a specific item

```
contents_bytes = client.drive_download_contents(item_id
```

Get a Drive item resource

```
drive_item_dict = client.drive_get_item(item_id)
```

Get worksheets

```
get_worksheets = client.excel_get_worksheets(item_id)
```

Get specific worksheet

```
specific_worksheet = client.excel_get_specific_workshee
```

Add worksheets

```
add_worksheet = client.excel_add_worksheet(item_id)
```

Update worksheet

```
update_worksheet = client.excel_update_worksheet(item_i
```

Get charts

```
get_charts = client.excel_get_charts(item_id, worksheet
```

Add chart

```
add_chart = client.excel_add_chart(item_id, worksheet_i
```

Get tables

```
get_tables = client.excel_get_tables(item_id)
```

Add table

```
add_table = client.excel_add_table(item_id)
```

Add column to table

```
add_column = client.excel_add_column(item_id, worksheet
```

Add row to table

```
add_row = client.excel_add_row(item_id, worksheets_id,
```

Get table rows

```
get_rows = client.excel_get_rows(item_id, table_id)
```

Get range

```
get_range = client.excel_get_range(item_id, worksheets_
```

Update range

```
update_range = client.excel_update_range(item_id, works
```

Requirements

- requests

Tests

```
test/test.py
```

Help

Installing packages
Uploading packages
User guide
FAQs

About PyPI

PyPI on Twitter
Infrastructure dashboard
Package index name retention
Our sponsors

Contributing to PyPI

Bugs and feedback
Contribute on GitHub
Translate PyPI
Development credits

Using PyPI

Code of conduct
Report security issue
Privacy policy
Terms of use

Status: All Systems Operational

Developed and maintained by the Python community, for the Python community.
Donate today!

Site map

Switch to desktop version

› English español français 日本語 Português Brasileiro Українська Ελληνικά Deutsch 简体中文 Русский

Pingdom
Monitoring

Google
Object Storage and Download Analytics

Sentry
Error logging

AWS
Cloud computing

DataDog
Monitoring