pyOutlook Documentation

Release 4.2.3

Jens Astrup

Oct 19, 2019

Contents

1 About: **3**
1.1 Python Versions 3

2 Recommended: **5**

3 Notes: **7**

4 Contents: **9**
4.1 Installation 9
4.2 Quick Start 9
4.3 pyOutlook 11

5 Indices and tables **19**

Index **21**

CHAPTER 1

About:

pyOutlook was created after I found myself attempting to connect to the Outlook REST API in multiple projects. This provided some much needed uniformity. It's easier to deal with than the win32com package by Microsoft, but obviously has a far smaller scope.

1.1 Python Versions

pyOutlook is only tested in, and targets, Python 2.7, 3.5, 3.6, 3.7, and 3.8.

Warning: Support for Python 2.7 will be dropped in version 5.0.0

CHAPTER 2

Recommended:

pyOutlook does not handle OAuth for the access tokens provided by Outlook. These are provided by you via the OutlookAccount class as a string. There are various OAuth packages out there: (pip install) oauth2, python-oauth2, requests_oauthlib, etc that can facilitate the process.

CHAPTER 3

Notes:

All protected & private methods and anything under pyOutlook.internal is subject to change without deprecation warnings

CHAPTER 4

Contents:

4.1 Installation

4.1.1 Pip

```
pip install pyOutlook
```

4.1.2 Source

pyOutlook's PyPI page has a tar.gz and zip distribution for each release.

4.2 Quick Start

4.2.1 Instantiation

pyOutlook interacts with Outlook messages and folders through the class OutlookAccount(). The OutlookAccount acts as a gatekeeper to the other methods available, and stores the access token associated with an account.

Instantiation Example:

```
from pyOutlook import OutlookAccount
account_one = OutlookAccount('token 1')
account_two = OutlookAccount('token 2')
```

From here you can access any of the methods as documented in the *pyOutlook* section. Here are two examples of accessing an inbox and sending a new email.

4.2.2 Examples

Retrieving Emails

Through the OutlookAccount class you can call one of many methods - `get_messages()`, `inbox()`, etc. These methods return a list of *Message* objects, allowing you to access the attributes therein.

```
inbox = account.inbox()
inbox[0].body
>>> 'A very fine body'
```

Sending Emails

As above, you can send emails through the OutlookAccount class. There are two methods for sending emails - one allows chaining of methods and the other takes all arguments upfront and immediately sends.

Message

You can create an instance of a `Message` and then send from there.

```
from pyOutlook import *
# or from pyOutlook.core.message import Message

account = OutlookAccount('token')
message = Message(account, 'A body', 'A subject', [Contact('to@email.com')])
message.attach(bytes('some bytes', 'utf-8'), 'bytes.txt')
message.send()
```

new_email()

This returns a `Message` instance.

```
body = 'I\'m sending an email through Python. <br> Best, <br>Me'

email = account.new_email(body=body, subject='Hey there', to=Contact('myemail@domain.
↪com'))
email.sender = Contact('some_other_account@email.com')
email.send()
```

Note that HTML formatting is accepted in the message body.

send_email()

This method takes all of its arguments at once and then sends.

```
account_one.send_email(
    to=[Contact('myemail@domain.com')],
    # or to=['myemail@domain.com')]
    subject='Hey there',
    body="I'm sending an email through Python. <br> Best, <br> Me",
)
```

Contacts

All recipients, and the sender attribute, in *Messages* are represented by *Contacts*. Right now, this allows you to retrieve the name of a recipient, if provided by Outlook.

```
message = account.inbox()[0]
message.sender.name
>>> 'Dude'
```

When providing recipients to *Message* you can provide them either as a list of strings, or a list of *Contacts*. I prefer the latter, as there are further options in the Outlook API for interacting with Contacts - functionality for those may be added in the future.

4.3 pyOutlook

4.3.1 Outlook Account

class pyOutlook.core.main.**OutlookAccount**(*access_token*)

Sets up access to Outlook account for all methods & classes.

access_token

A string OAuth token from Outlook allowing access to a user's account

class AutoReplyAudience

ALL = 'All'

CONTACTS_ONLY = 'ContactsOnly'

INTERNAL_ONLY = 'None'

class AutoReplyStatus

ALWAYS_ENABLED = 'AlwaysEnabled'

DISABLED = 'Disabled'

SCHEDULED = 'Scheduled'

auto_reply_message

The account's Internal auto reply message. Setting the value will change the auto reply message of the account, automatically setting the status to enabled (but not altering the schedule).

contact_overrides

deleted_messages()

last ten deleted messages.

Returns List[*Message*]

draft_messages()

last ten draft messages.

Returns List[*Message*]

get_folder_by_id(*folder_id*)

Retrieve a Folder by its Outlook ID

Parameters **folder_id** – The ID of the *Folder* to retrieve

Returns: *Folder*

get_folders()

Returns a list of all folders for this account

Returns List[*Folder*]

get_message(*message_id*)

Gets message matching provided id.

the Outlook email matching the provided message_id.

Parameters **message_id** – A string for the intended message, provided by Outlook

Returns *Message*

get_messages(*page=0*)

Get first 10 messages in account, across all folders.

Keyword Arguments **page** (*int*) – Integer representing the 'page' of results to fetch

Returns List[*Message*]

inbox()

first ten messages in account's inbox.

Returns List[*Message*]

new_email(*body=''*, *subject=''*, *to=<class 'list'>*)

Creates a *Message* object.

Keyword Arguments

- **body** (*str*) – The body of the email
- **subject** (*str*) – The subject of the email
- **to** (*List[Contact]*) – A list of recipients to email

Returns *Message*

send_email(*body=None*, *subject=None*, *to=<class 'list'>*, *cc=None*, *bcc=None*, *send_as=None*, *attachments=None*)

Sends an email in one method, a shortcut for creating an instance of *Message* .

Parameters

- **body** (*str*) – The body of the email
- **subject** (*str*) – The subject of the email
- **to** (*list*) – A list of *Contacts*
- **cc** (*list*) – A list of *Contacts* which will be added to the 'Carbon Copy' line
- **bcc** (*list*) – A list of *Contacts* while be blindly added to the email
- **send_as** (Contact) – A *Contact* whose email the OutlookAccount has access to
- **attachments** (*list*) – A list of dictionaries with two parts [1] 'name' - a string which will become the file's name [2] 'bytes' - the bytes of the file.

sent_messages()

last ten sent messages.

Returns List[*Message*]

set_auto_reply(*message*, *status='AlwaysEnabled'*, *start=None*, *end=None*, *external_message=None*, *audience='All'*)

Set an automatic reply for the account. :param message: The message to be sent in replies. If external_message is provided this is the message sent :type message: str :param to internal recipients: :param status: Whether the auto-reply should be always enabled, scheduled, or :type status: OutlookAccount.AutoReplyStatus :param disabled. You can use `AutoReplyStatus` to: :param provide the value. Defaults to ALWAYS_ENABLED.: :param start: If status is set to SCHEDULED, this is when the replies will start being sent. :type start: datetime :param end: If status is set to SCHEDULED, this is when the replies will stop being sent. :type end: datetime :param external_message: If provided, this message will be sent to external recipients. :type external_message: str :param audience: Whether replies should be sent to everyone, contacts only, :type audience: OutlookAccount.AutoReplyAudience :param or internal recipients only. You can use: :param `AutoReplyAudience` to provide the value.:

4.3.2 Message

class pyOutlook.core.message.**Message**(*account*, *body*, *subject*, *to_recipients*, *sender=None*, *cc=None*, *bcc=None*, *message_id=None*, ***kwargs*)

An object representing an email inside of the OutlookAccount.

message_id

A string provided by Outlook identifying this specific email

body

The body content of the email, including HTML formatting

body_preview

"The first 255 characters of the body"

subject

The subject of the email

sender

The `Contact` who sent this email. You can set this before sending an email to change which account the email comes from (so long as the `OutlookAccount` specified has access to the email.

to

A list of `Contacts`. You can also provide a list of strings, however these will be turned into `Contacts` after sending the email.

cc

A list of `Contacts` in the CC field. You can also provide a list of strings, however these will be turned into `Contacts` after sending the email.

bcc

A list of `Contacts` in the BCC field. You can also provide a list of strings, however these will be turned into `Contacts` after sending the email.

is_draft

Whether or not the email is a draft.

importance

The importance level of the email; with 0 indicating low, 1 indicating normal, and 2 indicating high. `Message.IMPORTANCE_LOW`, `Message.IMPORTANCE_NORMAL`, & `Message.IMPORTANCE_HIGH` can be used to reference the levels.

categories

A list of strings, where each string is the name of a category.

time_created
A datetime representing the time the email was created

time_sent
A datetime representing the time the email was sent

IMPORTANCE_HIGH = 2

IMPORTANCE_LOW = 0

IMPORTANCE_NORMAL = 1

add_category(*category_name*)

api_representation(*content_type*)
Returns the JSON representation of this message required for making requests to the API.

Parameters **content_type** (*str*) – Either 'HTML' or 'Text'

attach(*file_bytes*, *file_name*)
Adds an attachment to the email. The filename is passed through Django's get_valid_filename which removes invalid characters. From the documentation for that function:

```
>>> get_valid_filename("john's portrait in 2004.jpg")
'johns_portrait_in_2004.jpg'
```

Parameters

- **file_bytes** – The bytes of the file to send (if you send a string, ex for CSV, pyOutlook will attempt to convert that into bytes before base64ing the content).
- **file_name** – The name of the file, as a string and leaving out the extension, that should be sent

attachments

copy_to(*folder_id*)
Copies the email to the folder specified by the folder_id.

The folder id must match the id provided by Outlook.

Parameters **folder_id** – A string containing the folder ID the message should be copied to

copy_to_deleted()
Copies Message to account's Deleted Items folder

copy_to_drafts()
Copies Message to account's Drafts folder

copy_to_inbox()
Copies Message to account's Inbox

delete()
Deletes the email

focused
Sets and retrieves the 'Focused' status of a Message. If a user has the 'Focused' inbox, messages are divided between the Focused and Other tabs.

forward(*to_recipients*, *forward_comment=None*)
Forward Message to recipients with an optional comment.

Parameters

- **to_recipients** – A list of *Contacts* to send the email to.
- **forward_comment** – String comment to append to forwarded email.

Examples

```
>>> john = Contact('john.doe@domain.com')
>>> betsy = Contact('betsy.donalds@domain.com')
>>> email = Message()
>>> email.forward([john, betsy])
>>> email.forward([john], 'Hey John')
```

is_read

Set and retrieve the 'Read' status of an email

```
>>> message = Message()
>>> message.is_read
>>> False
>>> message.is_read = True
```

move_to(*folder*)

Moves the email to the folder specified by the folder parameter.

Parameters **folder** – A string containing the folder ID the message should be moved to, or a Folder instance

move_to_deleted()

Moves the email to the account's Deleted Items folder

move_to_drafts()

Moves the email to the account's Drafts folder

move_to_inbox()

Moves the email to the account's Inbox

parent_folder

Returns the *Folder* this message is in

```
>>> from pyOutlook import *
>>> account = OutlookAccount('')
>>> message = account.get_messages()[0]
>>> message.parent_folder
Inbox
>>> message.parent_folder.unread_count
19
```

Returns: *Folder*

reply(*reply_comment*)

Reply to the Message.

Notes

HTML can be inserted in the string and will be interpreted properly by Outlook.

Parameters **reply_comment** – String message to send with email.

reply_all(*reply_comment*)
Replies to everyone on the email, including those on the CC line.

With great power, comes great responsibility.

Parameters **reply_comment** – The string comment to send to everyone on the email.

send(*content_type='HTML'*)
Takes the recipients, body, and attachments of the Message and sends.

Parameters **content_type** – Can either be 'HTML' or 'Text', defaults to HTML.

4.3.3 Folder

class pyOutlook.core.folder.**Folder**(*account*, *folder_id*, *folder_name*, *parent_id*, *child_folder_count*, *unread_count*, *total_items*)
An object representing a Folder in the OutlookAccount provided.

account
The `OutlookAccount` this folder should be associated with

id
The static id generated by Outlook to identify this folder.

folder_name
The name of this folder as displayed in the account

parent_id
The id of the folder which houses this Folder object

child_folder_count
The number of child folders inside this Folder

unread_count
The number of unread messages inside this Folder

total_items
A sum of all items inside Folder

> **Warning:** Setting the id attribute on a folder instance will no longer be supported in v5.0.0

copy_into(*destination_folder*)
Copies the Folder into the provided destination folder.

Raises `AuthError` – Raised if Outlook returns a 401, generally caused by an invalid or expired access token.

Parameters **destination_folder** – The Folder that this Folder should be copied to.

Returns A new `Folder` representing the newly created folder.

create_child_folder(*folder_name*)
Creates a child folder within the Folder it is called from and returns the new Folder object.

Parameters **folder_name** – The name of the folder to create

Returns: `Folder`

delete()
Deletes this Folder.

Raises `AuthError` – Raised if Outlook returns a 401, generally caused by an invalid or expired access token.

get_subfolders()

Retrieve all child Folders inside of this Folder.

Raises `AuthError` – Raised if Outlook returns a 401, generally caused by an invalid or expired access token.

Returns List[*Folder*]

messages()

Retrieves the messages in this Folder, returning a list of *Messages*.

move_into(*destination_folder*)

Move the Folder into a different folder.

This makes the Folder provided a child folder of the destination_folder.

Raises `AuthError` – Raised if Outlook returns a 401, generally caused by an invalid or expired access token.

Parameters **destination_folder** – A *Folder* that should become the parent

Returns A new *Folder* that is now inside of the destination_folder.

rename(*new_folder_name*)

Renames the Folder to the provided name.

Parameters **new_folder_name** – A string of the replacement name.

Raises `AuthError` – Raised if Outlook returns a 401, generally caused by an invalid or expired access token.

Returns A new Folder representing the folder with the new name on Outlook.

4.3.4 Contact

class `pyOutlook.core.contact.`**Contact**(*email*, *name=None*, *focused=None*)

Represents someone sending or receiving an email. Cuts down on the amount of dictionaries floating around that each hold the API's syntax and allows for functionality to be added in the future.

Parameters

- **email** (*str*) – The email of the user
- **name** (*str*) – The user's name, which is not always provided by the API.

Keyword Arguments **focused** – Whether messages from this sender are always sent to the Focused inbox, or to the Other tab. This value is set when retrieving a contact from the API, or after setting it via *set_focused()*

email

The email of the user

name

The name of the user

focused

A boolean indicating whether this contact has an override for their messages to go to the Focused inbox or the "Other" inbox. None indicates that the value has not yet been retrieved by the API or set.

api_representation()

Returns the JSON formatting required by Outlook's API for contacts

set_focused(*account*, *is_focused*)

Emails from this contact will either always be put in the Focused inbox, or always put in Other, based on the value of is_focused.

Parameters

- **account** (`OutlookAccount`) – The *`OutlookAccount`* the override should be set for
- **is_focused** (*`bool`*) – Whether this contact should be set to Focused, or Other.

Returns True if the request was successful

CHAPTER 5

Indices and tables

- genindex
- search

Index

A

access_token (*pyOutlook.core.main.OutlookAccount attribute*), 11
account (*pyOutlook.core.folder.Folder attribute*), 16
add_category() (*pyOutlook.core.message.Message method*), 14
ALL (*pyOutlook.core.main.OutlookAccount.AutoReplyAudience attribute*), 11
ALWAYS_ENABLED (*pyOutlook.core.main.OutlookAccount.AutoReplyStatus attribute*), 11
api_representation() (*pyOutlook.core.contact.Contact method*), 17
api_representation() (*pyOutlook.core.message.Message method*), 14
attach() (*pyOutlook.core.message.Message method*), 14
attachments (*pyOutlook.core.message.Message attribute*), 14
auto_reply_message (*pyOutlook.core.main.OutlookAccount attribute*), 11

B

bcc (*pyOutlook.core.message.Message attribute*), 13
body (*pyOutlook.core.message.Message attribute*), 13
body_preview (*pyOutlook.core.message.Message attribute*), 13

C

categories (*pyOutlook.core.message.Message attribute*), 13
cc (*pyOutlook.core.message.Message attribute*), 13
child_folder_count (*pyOutlook.core.folder.Folder attribute*), 16
Contact (*class in pyOutlook.core.contact*), 17
contact_overrides (*pyOutlook.core.main.OutlookAccount attribute*), 11
CONTACTS_ONLY (*pyOutlook.core.main.OutlookAccount.AutoReplyAudience attribute*), 11
copy_into() (*pyOutlook.core.folder.Folder method*), 16
copy_to() (*pyOutlook.core.message.Message method*), 14
copy_to_deleted() (*pyOutlook.core.message.Message method*), 14
copy_to_drafts() (*pyOutlook.core.message.Message method*), 14
copy_to_inbox() (*pyOutlook.core.message.Message method*), 14
create_child_folder() (*pyOutlook.core.folder.Folder method*), 16

D

delete() (*pyOutlook.core.folder.Folder method*), 16
delete() (*pyOutlook.core.message.Message method*), 14
deleted_messages() (*pyOutlook.core.main.OutlookAccount method*), 11
DISABLED (*pyOutlook.core.main.OutlookAccount.AutoReplyStatus attribute*), 11
draft_messages() (*pyOutlook.core.main.OutlookAccount method*), 11

E

email (*pyOutlook.core.contact.Contact attribute*), 17

F

focused (*pyOutlook.core.contact.Contact attribute*), 17
focused (*pyOutlook.core.message.Message attribute*), 14
Folder (*class in pyOutlook.core.folder*), 16
folder_name (*pyOutlook.core.folder.Folder attribute*), 16

forward() (*pyOutlook.core.message.Message method*), 14

G

get_folder_by_id() (*pyOutlook.core.main.OutlookAccount method*), 11
get_folders() (*pyOutlook.core.main.OutlookAccount method*), 12
get_message() (*pyOutlook.core.main.OutlookAccount method*), 12
get_messages() (*pyOutlook.core.main.OutlookAccount method*), 12
get_subfolders() (*pyOutlook.core.folder.Folder method*), 17

I

id (*pyOutlook.core.folder.Folder attribute*), 16
importance (*pyOutlook.core.message.Message attribute*), 13
IMPORTANCE_HIGH (*pyOutlook.core.message.Message attribute*), 14
IMPORTANCE_LOW (*pyOutlook.core.message.Message attribute*), 14
IMPORTANCE_NORMAL (*pyOutlook.core.message.Message attribute*), 14
inbox() (*pyOutlook.core.main.OutlookAccount method*), 12
INTERNAL_ONLY (*pyOutlook.core.main.OutlookAccount.AutoReplyAudience attribute*), 11
is_draft (*pyOutlook.core.message.Message attribute*), 13
is_read (*pyOutlook.core.message.Message attribute*), 15

M

Message (*class in pyOutlook.core.message*), 13
message_id (*pyOutlook.core.message.Message attribute*), 13
messages() (*pyOutlook.core.folder.Folder method*), 17
move_into() (*pyOutlook.core.folder.Folder method*), 17
move_to() (*pyOutlook.core.message.Message method*), 15
move_to_deleted() (*pyOutlook.core.message.Message method*), 15
move_to_drafts() (*pyOutlook.core.message.Message method*), 15
move_to_inbox() (*pyOutlook.core.message.Message method*), 15

N

name (*pyOutlook.core.contact.Contact attribute*), 17
new_email() (*pyOutlook.core.main.OutlookAccount method*), 12

O

OutlookAccount (*class in pyOutlook.core.main*), 11
OutlookAccount.AutoReplyAudience (*class in pyOutlook.core.main*), 11
OutlookAccount.AutoReplyStatus (*class in pyOutlook.core.main*), 11

P

parent_folder (*pyOutlook.core.message.Message attribute*), 15
parent_id (*pyOutlook.core.folder.Folder attribute*), 16

R

rename() (*pyOutlook.core.folder.Folder method*), 17
reply() (*pyOutlook.core.message.Message method*), 15
reply_all() (*pyOutlook.core.message.Message method*), 15

S

SCHEDULED (*pyOutlook.core.main.OutlookAccount.AutoReplyStatus attribute*), 11
send() (*pyOutlook.core.message.Message method*), 16
send_email() (*pyOutlook.core.main.OutlookAccount method*), 12
sender (*pyOutlook.core.message.Message attribute*), 13
sent_messages() (*pyOutlook.core.main.OutlookAccount method*), 12
set_auto_reply() (*pyOutlook.core.main.OutlookAccount method*), 12
set_focused() (*pyOutlook.core.contact.Contact method*), 18
subject (*pyOutlook.core.message.Message attribute*), 13

T

time_created (*pyOutlook.core.message.Message attribute*), 13
time_sent (*pyOutlook.core.message.Message attribute*), 14
to (*pyOutlook.core.message.Message attribute*), 13
total_items (*pyOutlook.core.folder.Folder attribute*), 16

U

`unread_count` (*pyOutlook.core.folder.Folder attribute*), 16