Mail Sender SU

Version 0.2.0

Quentin Deschamps

janv. 30, 2024

Plan

1 Installation **3**
- 1.1 Avec pip . . . 3
- 1.2 Avec setuptools . . . 3
- 1.3 Sur Windows . . . 3

2 Utilisation **5**
- 2.1 GUI . . . 6
  - 2.1.1 Authentification . . . 6
  - 2.1.2 Votre adresse . . . 7
  - 2.1.3 Destinataires . . . 7
  - 2.1.4 Pièces jointes . . . 8
  - 2.1.5 Envoyer . . . 8
  - 2.1.6 Option style . . . 8
- 2.2 CLI . . . 9
  - 2.2.1 Envoyer un mail . . . 9
  - 2.2.2 Gérer les adresses . . . 9
  - 2.2.3 Gérer la signature . . . 10

3 API **11**
- 3.1 Emails utils . . . 11
- 3.2 Files . . . 12
- 3.3 Login interface . . . 12
- 3.4 Mail interface . . . 13
- 3.5 Attachments . . . 13
- 3.6 Style . . . 13
- 3.7 Message . . . 13

4 Liens **15**

5 Auteur **17**

6 Index **19**

Index des modules Python **21**

Index **23**

Mail Sender SU est une application de mail sender en Python pour les étudiants de Sorbonne-Université. Une CLI (*Command Line Interface*) associée à une GUI (*Graphical User Interface*) permettent d'envoyer des mails rapidement et facilement avec son adresse *sorbonne-universite*.

CHAPITRE 1

Installation

1.1 Avec pip

Mail Sender SU est installable avec la commande `pip3` :

```
pip3 install mailSenderSU
```

1.2 Avec setuptools

Une autre manière d'installer l'application est de cloner le repository GitHub et d'utiliser *setuptools* :

```
git clone https://github.com/Quentin18/Mail-Sender-SU.git
python3 setup.py install
```

1.3 Sur Windows

En installant l'application sur Windows, vous verrez peut-être apparaître le warning suivant :

```
The script mailSenderSU.exe is installed in 'C:\Users\...' which is not on PATH.
```

Pour corriger ce warning et pouvoir utiliser la commande `mailSenderSU`, vous devez ajouter le chemin affiché par le message dans le PATH. Vous pouvez suivre le tutoriel ici.

CHAPITRE 2

Utilisation

Cette partie explique comment utiliser l'application *Mail Sender SU*.

Pour afficher l'ensemble des commandes de l'application, entrez :

```
mailSenderSU --help
```

Vous devez obtenir l'affichage suivant :

```
Usage: mailSenderSU [OPTIONS] COMMAND [ARGS]...

Mail Sender Sorbonne-University.

Options:
-v, --verbosity  Set the verbosity
--help           Show this message and exit.

Commands:
addcontact   Add contact ADDRESS.
adduser      Add user ADDRESS.
clear        Clear all data files.
contacts     Edit contacts.
delcontacts  Delete contacts.
deluser      Delete user addresses.
gui          Graphical user interface.
send         Send mail.
signature    Edit signature.
user         Edit user.
```

2.1 GUI

Pour utiliser la GUI, utilisez la commande :

```
mailSenderSU gui
```

La fenêtre suivante doit apparaître :



2.1.1 Authentification

Entrez votre numéro étudiant et votre mot de passe pour vous connecter au serveur. Si la connexion est établie, vous êtes dirigés sur la fenêtre suivante.



2.1.2 Votre adresse

— Dans le premier champ, vous devez utiliser votre adresse *sorbonne-universite.fr* ou *upmc.fr*. Une adresse externe à l'université ne fonctionnera pas.

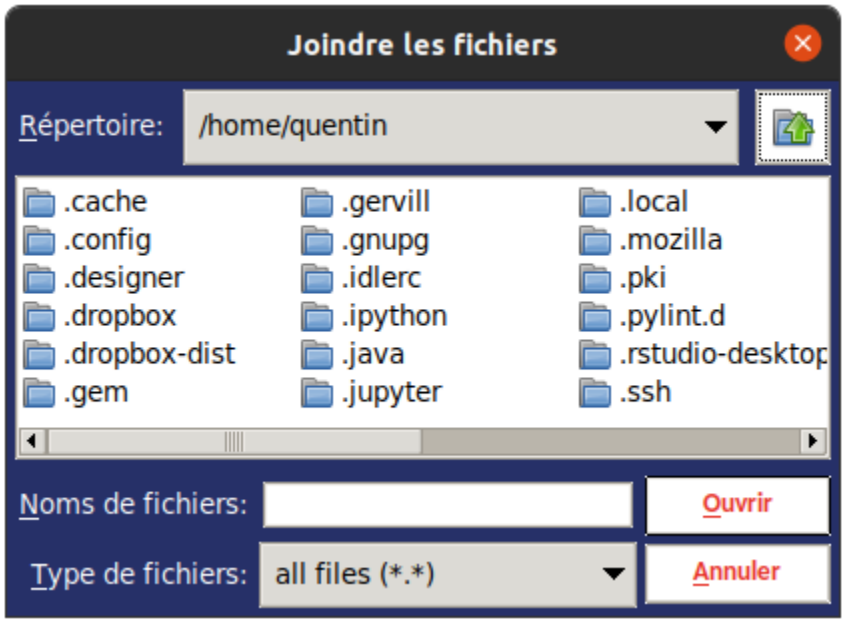
2.1.3 Destinataires

— Dans le deuxième champ, vous devez mettre les adresses des destinataires.
— Dans le troisième champ, se sont les destinataires en copie.

Note : Veuillez séparer les adresses par **des virgules** s'il y en a plusieurs.

2.1.4 Pièces jointes

Pour insérer des pièces jointes, pressez le bouton *Joindre un fichier*. La fenêtre suivante doit apparaître :



Sélectionner le(s) fichier(s) à insérer. Il apparaîtront dans le cadre du bas. Vous pouvez supprimer une pièce jointe en cliquant sur celle-ci et en pressant le bouton *Supprimer*.

Note : Tout type de fichier peut être envoyé, même les archives *.zip* ou *.tar.gz*.

2.1.5 Envoyer

Cliquez sur le bouton *Envoyer* pour lancer l'envoi. Un message vous indiquera si l'envoi a réussi ou échoué. On vous proposera aussi d'ajouter les destinataires à vos contacts si ceux-ci n'y sont pas.

2.1.6 Option style

Avec l'option `--style`, vous pouvez choisir le style de la fenêtre entre `su` ou `polytech` :

```
mailSenderSU gui --style polytech
```

Voici le résultat de la commande ci-dessus :



2.2 CLI

Voici les différentes commandes de la CLI :

2.2.1 Envoyer un mail

Il est aussi possible d'envoyer un mail via le terminal avec la commande `send` :

```
mailSenderSU send [OPTIONS]
```

On vous demandera :
- Votre identifiant (numéro étudiant)
- Votre mot de passe
- Votre adresse d'envoi
- Les adresses des destinataires (et ceux en copie)
- L'objet du mail
- Le message
- Les pièces jointes

Note : Cette méthode est à utiliser si le message est plutôt court. Sinon, privilégier l'utilisation de la GUI.

2.2.2 Gérer les adresses

L'application *Mail Sender SU* garde en mémoire vos adresses et celles de vos contacts. Vous pouvez gérer ces adresses avec les commandes suivantes :
- `addcontact` : ajoute une nouvelle adresse à votre liste des contacts.

```
mailSenderSU addcontact [OPTIONS] ADDRESS
```

- `adduser` : ajoute une nouvelle adresse utilisateur.

```
mailSenderSU adduser [OPTIONS] ADDRESS
```

- `contacts` : ouvre le fichier des contacts sur votre éditeur de texte par défaut. Vous pouvez alors ajouter ou supprimer des contacts facilement.

```
mailSenderSU contacts [OPTIONS]
```

— `user` : ouvre le fichier des adresses utilisateur sur votre éditeur de texte par défaut.

```
mailSenderSU user [OPTIONS]
```

— `delcontacts` : supprime tous vos contacts.

```
mailSenderSU delcontacts [OPTIONS]
```

— `deluser` : supprime toutes vos adresses utilisateur.

```
mailSenderSU deluser [OPTIONS]
```

— `clear` : supprime vos adresses et vos contacts.

```
mailSenderSU clear [OPTIONS]
```

Note : Un groupe de plusieurs adresses peut être inscrit dans les contacts (sur une ligne, séparer les adresses par des virgules).

2.2.3 Gérer la signature

Les mails envoyés avec *Mail Sender SU* contiennent une signature automatique avec votre nom et les logos de l'université comme ci-dessous.



Vous pouvez modifier cette signature HTML avec la commande suivante :

```
mailSenderSU signature [OPTIONS]
```

Le fichier `signature.html` utilisé sera ouvert dans votre éditeur de texte par défaut et vous pourrez ajouter des informations, des liens ou des images.

CHAPITRE 3

API

Cette partie contient les modules de la librairie `mailSenderSU`. Ils sont contenus dans le répertoire `src`.

Importation :

```
import mailSenderSU.src
```

3.1 Emails utils

The `email_utils` module is used to create and send emails. It uses `smtplib` to send an email.

class email_utils.**Email**(*from_*, *to*, *cc*, *subject*, *message*, *attachments*, *signature*)

Bases : `object`

Manage emails.

static get_name_from_address(*address*)

Return the name associated to the address.

static add_name_to_signature(*signature*, *name*, *color='#263068'*)

Add the name to the signature.

class email_utils.**EmailConnection**(*username*, *password*, *server='smtps.upmc.fr'*, *port=587*)

Bases : `object`

Manage communication with the SMTP server.

connect()

Connect to the server.

static is_valid(*username*, *password*, *server='smtps.upmc.fr'*, *port=587*)

Return True if the username and the password are valid.

static get_list(*addresses*)

Return a list of addresses from a string.

send(*message*)

Send an email.

close()
Close connection.

3.2 Files

The `files` module manages files. The `File` class can edit a file.

class files.**File**(*filename*)
Bases : `object`
Manage a file.
create()
Create an empty file.
property filename
Get filename.
read()
Read a file.
get_list()
Return the list from a file.
add(*data*)
Add data to a file.
clear()
Delete all data from a file.

3.3 Login interface

The `login_interface` module creates the login interface of the GUI.

class login_interface.**LoginInterface**(*window*, *style*)
Bases : `object`
Manage login interface.
property login
Get login.
property username
Get username.
property password
Get password.
return_connect(*event*)
Connect to the server with the <ENTER> command.
connect()
Try to connect to the server.

3.4 Mail interface

The `mail_interface` module creates the mail sender interface of the GUI.

class `mail_interface.`**MailSenderInterface**(*window*, *style*, *username*, *password*)

Bases : `object`

Manage mail sender interface.

property username

Get username.

property password

Get password.

check()

Check fields.

send()

Send an email.

3.5 Attachments

The `attachments` module manages attachments. It uses the `askopenfilenames` function from the `tkinter.filedialog` module.

class `attachments.`**Attachments**(*window*, *button_frame*)

Bases : `object`

Manage attachments.

attach_files()

Add attachments.

delete_file()

Delete attachments.

3.6 Style

The `style` module manages the style of the GUI.

class `style.`**Style**(*window*, *style*)

Bases : `object`

Manage the style of a window.

3.7 Message

The `message` module manages the messages of the GUI. It uses the `tkinter.messagebox` module.

`message.`**show_login_error**()

Login error message.

`message.`**show_send**()

Success send message.

message.**show_from_miss**()
Miss “from” field message.

message.**show_to_miss**()
Miss “to” field message.

message.**show_address_not_valid**(*address*)
“Not valid address” message.

message.**show_subject_miss**()
Miss “subject” field message.

message.**show_new_user**(*address*)
New user message.

message.**show_new_contact**(*address*)
New contact message.

message.**show_send_error**()
Send error message.

CHAPITRE 4

Liens

— GitHub : https://github.com/Quentin18/Mail-Sender-Sorbonne-Universite/
— PyPI : https://pypi.org/project/mailSenderSU/
— Documentation : https://mail-sender-sorbonne-universite.readthedocs.io/fr/latest/
— Travis : https://travis-ci.org/github/Quentin18/Mail-Sender-Sorbonne-Universite/

CHAPITRE 5

Auteur

Quentin Deschamps : quentin.deschamps@etu.sorbonne-universite.fr

CHAPITRE 6

Index

— genindex
— modindex
— search



Index des modules Python

a

attachments, 13

e

email_utils, 11

f

files, 12

l

login_interface, 12

m

mail_interface, 13
message, 13

s

style, 13

Index

A

add() (*méthode files.File*), 12
add_name_to_signature() (*méthode statique email_utils.Email*), 11
attach_files() (*méthode attachments.Attachments*), 13
attachments
    module, 13
Attachments (*classe dans attachments*), 13

C

check() (*méthode mail_interface.MailSenderInterface*), 13
clear() (*méthode files.File*), 12
close() (*méthode email_utils.EmailConnection*), 11
connect() (*méthode email_utils.EmailConnection*), 11
connect() (*méthode login_interface.LoginInterface*), 12
create() (*méthode files.File*), 12

D

delete_file() (*méthode attachments.Attachments*), 13

E

Email (*classe dans email_utils*), 11
email_utils
    module, 11
EmailConnection (*classe dans email_utils*), 11

F

File (*classe dans files*), 12
filename (*propriété files.File*), 12
files
    module, 12

G

get_list() (*méthode files.File*), 12
get_list() (*méthode statique email_utils.EmailConnection*), 11
get_name_from_address() (*méthode statique email_utils.Email*), 11

I

is_valid() (*méthode statique email_utils.EmailConnection*), 11

L

login (*propriété login_interface.LoginInterface*), 12
login_interface
    module, 12
LoginInterface (*classe dans login_interface*), 12

M

mail_interface
    module, 13
MailSenderInterface (*classe dans mail_interface*), 13
message
    module, 13
module
    attachments, 13
    email_utils, 11
    files, 12
    login_interface, 12
    mail_interface, 13
    message, 13
    style, 13

P

password (*propriété login_interface.LoginInterface*), 12
password (*propriété mail_interface.MailSenderInterface*), 13

R

read() (*méthode files.File*), 12
return_connect() (*méthode login_interface.LoginInterface*), 12

S

send() (*méthode email_utils.EmailConnection*), 11

send() (*méthode mail_interface.MailSenderInterface*), 13
show_address_not_valid() (*dans le module message*), 14
show_from_miss() (*dans le module message*), 13
show_login_error() (*dans le module message*), 13
show_new_contact() (*dans le module message*), 14
show_new_user() (*dans le module message*), 14
show_send() (*dans le module message*), 13
show_send_error() (*dans le module message*), 14
show_subject_miss() (*dans le module message*), 14
show_to_miss() (*dans le module message*), 14
style
    module, 13
Style (*classe dans style*), 13

U

username (*propriété login_interface.LoginInterface*), 12
username (*propriété mail_interface.MailSenderInterface*), 13