emaillib Documentation

Release 0.1.0

Costas Tyfoxylos

Sep 16, 2017

Contents

Contents:

CHAPTER 1

emaillib

A simple library to ease the sending of emails with utf8 and attachments.

- Documentation: http://emaillib.readthedocs.io/en/latest

Features

- Handles subject and message body as utf-8 by default
- Handles html and plain text
- Handles attachments either as list or comma delimited value arguments
- Supports CC and BCC
- Simply validates the provided recipient addresses

CHAPTER 2

Installation

At the command line:

```
$ pip install emaillib
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv emaillib
$ pip install emaillib
```

CHAPTER 3

Usage

To use emaillib in a project:

```
# to easily just send a message
from emaillib import EasySender

smtp = {'address': 'smtp.domain.com',
        'username': 'usersName',
        'password': 'usersPassword',
        'port': 587}
info = {'sender': 'test@test.com',
        'recipients': 'whatever@gmail.com',
        'cc': ['somebody@gmail.com'],
        'bcc': 'more@gmail.com,andmore@gmail.com',
        'subject': 'Τϵστ test',
        'body': 'This is a τϵστ on utf8'}

server = EasySender(**smtp)
server.send(**info)


# to use a more constant server connection where you can connect and
# disconnect as required

from emaillib import SmtpServer

smtp = {'address': 'smtp.domain.com',
        'username': 'usersName',
        'password': 'usersPassword',
        'port': 587}
info = {'sender': 'test@test.com',
        'recipients': 'whatever@gmail.com',
        'cc': ['somebody@gmail.com'],
        'bcc': 'more@gmail.com,andmore@gmail.com',
        'subject': 'Τϵστ test',
        'body': 'This is a τϵστ on utf8'}
```

```
server = SmtpServer(**smtp)
server.connect()
server.send(**info)
server.disconnect()


# a message can manually be constructed
# values for recipients, cc and bcc can be either lists or tuples, or comma
# delimited text. Internally those will be transformed to lists.
from emaillib import Message

info = {'sender': 'test@test.com',
        'recipients': 'whatever@gmail.com',
        'cc': ['somebody@gmail.com'],
        'bcc': 'more@gmail.com,andmore@gmail.com',
        'subject': 'Τεστ test',
        'body': 'This is a τεστ on utf8'}

message = Message(**info)

# show all the recipients
print(message.recipients)
# >>> ['whatever@gmail.com', 'somebody@gmail.com', 'more@gmail.com', 'andmore@gmail.
↪com']

# show only "to" recipients
print(message.to)
# >>> ['whatever@gmail.com']

# show only "cc" recipients
print(message.cc)
# >>> ['somebody@gmail.com']

# show only "bcc" recipients
print(message.bcc)
# >>> ['more@gmail.com', 'andmore@gmail.com']

# and its string representation can be accessed as
print(message.as_string)
```

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

Submit Feedback

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.

Get Started!

Ready to contribute? Here's how to set up *emaillib* for local development.

1. Clone your fork locally:

   ```
   $ git clone https://github.com/costastf/emaillib.git
   ```

2. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your clone for local development:

   ```
   $ mkvirtualenv emaillib
   $ cd emaillib/
   $ python setup.py develop
   ```

3. Create a branch for local development:

   ```
   $ git checkout -b name-of-your-bugfix-or-feature
   ```

   Now you can make your changes locally.

4. Commit your changes and push your branch to the server:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

5. Submit a merge request

CHAPTER 5

emaillib

emaillib package

Submodules

emaillib.emaillib module

Main module file

class `emaillib.emaillib.`**`EasySender`**(*address*, *username=None*, *password=None*, *tls=False*, *ssl=True*)

Bases: `object`

A simple wrapper around the SmtpServer object

`send`(*sender*, *recipients*, *cc=None*, *bcc=None*, *subject=''*, *body=''*, *attachments=None*, *content='text'*)

Sends the email by connecting and disconnecting after the send

Parameters

- **`sender`** – The sender of the message
- **`recipients`** – The recipients (To:) of the message
- **`cc`** – The CC recipients of the message
- **`bcc`** – The BCC recipients of the message
- **`subject`** – The subject of the message
- **`body`** – The body of the message
- **`attachments`** – The attachments of the message
- **`content`** – The type of content the message [text/html]

Returns True on success, False otherwise

class `emaillib.emaillib.`**`Message`**(*sender*, *recipients*, *cc=None*, *bcc=None*, *subject=''*, *body=''*, *attachments=None*, *content='text'*)

Bases: `object`

A model of an email message

`as_string`

The string representation of the message

`attachments`

A list of attachment names of the message

`bcc`

The bcc recipients of the message

`body`

The body of the message

`cc`

The cc recipients of the message

`content`

The type of content of the message

`recipients`

A list of all recipients of the message

`sender`

The email address of the sender

`subject`

The subject of the message

`to`

The main (to) recipients of the message

class `emaillib.emaillib.`**`SmtpServer`**(*address*, *username=None*, *password=None*, *tls=True*, *ssl=False*, *port=None*)

Bases: `object`

A simple wrapper around build in smtplib capabilities

`address`

The smtp server address upon instantiation

`connect`()

Initializes a connection to the smtp server

Returns True on success, False otherwise

`connected`

The status of connection to the smtp server

`disconnect`()

Disconnects from the remote smtp server

Returns True on success, False otherwise

`password`

The password upon instantiation

`port`

The smtp server port upon instantiation

send(*sender*, *recipients*, *cc=None*, *bcc=None*, *subject=''*, *body=''*, *attachments=None*, *content='text'*)
Sends the email

Parameters

- **sender** – The server of the message
- **recipients** – The recipients (To:) of the message
- **cc** – The CC recipients of the message
- **bcc** – The BCC recipients of the message
- **subject** – The subject of the message
- **body** – The body of the message
- **attachments** – The attachments of the message
- **content** – The type of content the message [text/html]

Returns True on success, False otherwise

ssl
The setting of ssl upon instantiation

tls
The setting of tls upon instantiation

username
The username upon instantiation

emaillib.emaillibexceptions module

Main module Exceptions file

Put your exception classes here

Module contents

emaillib package

Imports all parts from emaillib here

CHAPTER 6

Credits

Development Lead

- Costas Tyfoxylos <costas.tyf@gmail.com>

Contributors

None yet. Why not be the first?

CHAPTER 7

History

CHAPTER 8

0.1 (16-09-2017)

- First release

CHAPTER 9

Indices and tables

- genindex
- modindex
- search

x‘ * search

Python Module Index

Index