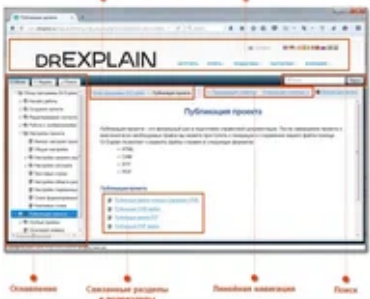


ХОЧУ ПОМОЧЬ  
ПРОЕКТО

Как отправлять электронные письма с помощью Python



drexpain.ru

РЕКЛАМА

Делать документацию в Dr.Explain быстрее, чем в MS Word

Бесплатная демо-версия • Русскоязычный интерфейс • Реестр отечественного ПО • Русскоязычная поддержка

Узнать больше

ChatGPT повысит уровень ваших продаж!

Больше не придется тратить время на прослушку звонков и контроль менеджеров!

Цены

Регистрация

Оставить заявку

Заказать звонок

Узнать больше

отправлять электронные письма с помощью Python

письма (email) с вложением

есть несколько отличных модулей, которые можно использовать для [создания и отправки](#) mail и smtp lib. В этом материале не будут рассматриваться API этих двух модулей дут представлены реальные приемы использования этих модулей.

т следующее:

[письмо с помощью smtp lib.](#)

[на несколько адресов.](#)

[почту, используя строки CC и BCC.](#)

[то письма с помощью модуля email.](#)

## Как отправить электронное письмо с помощью smtp lib.

Если посмотреть на API модуля smtp lib, то сразу становится понятно, как его использовать. Напишем быстрый пример, который показывает, как отправить электронное письмо.

```
import smtp lib

# данные почтового сервиса
user = "your-address@yandex.ru"
passwd = "*****"
server = "smtp.yandex.ru"
port = 587

# тема письма
subject = "Тестовое письмо от Python."
# кому
to = "person@mail.ru"
# кодировка письма
charset = 'Content-Type: text/plain; charset=utf-8'
mime = 'MIME-Version: 1.0'
# текст письма
text = "Отправкой почты управляет Python!"

# формируем тело письма
body = "\r\n".join((f"From: {user}", f"To: {to}",
                    f"Subject: {subject}", mime, charset, "", text))

try:
    # подключаемся к почтовому сервису
    smtp = smtp lib.SMTP(server, port)
    smtp.starttls()
    smtp.ehlo()
    # логинимся на почтовом сервере
    smtp.login(user, passwd)
    # будем послать письмо
    smtp.sendmail(user, to, body.encode('utf-8'))
```

Вверх

```
except smtplib.SMTPException as err:
    print('Что - то пошло не так...')
    raise err
finally:
```



new.sipuni.ru

### ChatGPT повысит уровень ваших продаж!

Больше не придется тратить время на прослушку звонков и контроль менеджеров!

Цены

Регистрация

Оставить заявку

Заказать звонок

Узнать больше

ль smtplib. Две трети этого кода используются для настройки электронной почты. ют пояснений, поэтому рассмотрим только одну переменную body.

пользуется метод [str.join\(\)](#), который объединяет все предыдущие переменные в одну чивается символом возврата каретки ('/r') плюс новая строка ('/n'). В итоге, ь как-то так:

```
r\nTo: person@mail.ru\r\n
thon.\r\nMIME-Version: 1.0\r\n
set=utf-8\r\n\r\n
on!'
```

единение с сервером электронной почты, а затем вызываете метод smtp.sendmail модуля й почты. Если сервер электронной почты требует аутентификации, то перед отправкой ься на этом сервере:

оборачивают в функцию и вызывают ее с некоторыми из этих параметров.

```
addr, subject, text, encode='utf-8'))
```

```
"""
Отправка электронного письма (email)
"""

# оставшиеся настройки
passwd = "*****"
server = "smtp.yandex.ru"
port = 587
charset = f'Content-Type: text/plain; charset={encode}'
mime = 'MIME-Version: 1.0'
# формируем тело письма
body = "\r\n".join((f"From: {from_addr}", f"To: {to_addr}",
                    f"Subject: {subject}", mime, charset, "", text))


try:
    # подключаемся к почтовому сервису
    smtp = smtplib.SMTP(server, port)
    smtp.starttls()
    smtp.ehlo()
    # логинимся на почтовом сервере
    smtp.login(from_addr, passwd)
    # пробуем послать письмо
    smtp.sendmail(from_addr, to_addr, body.encode(encode))
except smtplib.SMTPException as err:
    print('Что - то пошло не так...')
    raise err
finally:
    smtp.quit()


if __name__ == "__main__":
    from_addr = "your-address@yandex.ru"
    to_addr = "person@mail.ru"
    subject = "Тестовое письмо от Python."
    text = "Отправкой почты управляет Python!"
    send_email(from_addr, to_addr, subject, text)
```

Вверх

Иногда, часть оставшихся настроек задается файлом конфигурации в создаваемом приложения. Добавим файл конфигурации для хранения информации о сервере и адресе отправителя. Сохраним файл конфигурации как `email.ini`. Это удобно, если почтовый сервер обновляется и имя меняется, то не надо лезть в код, нужно только изменить файл конфигурации.

РЕКЛАМА



 new.sipuni.ru

### ChatGPT повысит уровень ваших продаж!

Больше не придется тратить время на прослушку звонков и контроль менеджеров!

Цены

>

Регистрация

>

Оставить заявку

>

Заказать звонок

>

Узнать больше

нфигурации, в котором есть только один раздел с пометкой `[smtp]`. Чтобы прочитать его в словарь Python, будем использовать [модуль ConfigParser](#). Вот обновленная версия

```
gParser

t, text, encode='utf-8'):

    ma (email)

e(os.path.abspath(__file__))
(base_path, "email.ini")
email.ini`
ath):

х из конфигурации

server = cfg.get("smtp", "server")
port = cfg.get("smtp", "port")
from_addr = cfg.get("smtp", "email")
passwd = cfg.get("smtp", "passwd")
else:
    print("Конфигурация не найдена!")
    sys.exit(1)

charset = f'Content-Type: text/plain; charset={encode}'
mime = 'MIME-Version: 1.0'
# формируем тело письма
body = "\r\n".join((f"From: {from_addr}", f"To: {to_addr}",
    f"Subject: {subject}", mime, charset, "", text))


try:
    # подключаемся к почтовому сервису
    smtp = smtplib.SMTP(server, port)
    smtp.starttls()
    smtp.ehlo()
    # логинимся на почтовом сервере
    smtp.login(from_addr, passwd)
    # пробуем послать письмо
    smtp.sendmail(from_addr, to_addr, body.encode(encode))
except smtplib.SMTPException as err:
    print('Что - то пошло не так...')
    raise err
finally:
    smtp.quit()

if __name__ == "__main__":
    to_addr = "person@mail.ru"
    subject = "Тестовое письмо от Python."
    text = "Отправкой почты управляет Python!"
    send_email(to_addr, subject, text)
```

Вверх

Добавили небольшую проверку в код. Сначала получаем путь, по которому находится сам скрипт отправки электронной почты `base_path`. Затем объединяем этот путь с именем `email.ini`, чтобы получить полный путь к файлу конфигурации. Потом проверяем наличие этого файла и в случае успешной проверки создаем объект [ConfigParser](#), если нет, то выводим сообщение и выходим из сценария.



new.sipuni.ru

### ChatGPT повысит уровень ваших продаж!

Больше не придется тратить время на прослушку звонков и контроль менеджеров!

Цены	>
Регистрация	>
Оставить заявку	>
Заказать звонок	>

Узнать больше

Добавим обработчик исключений вокруг вызова `ConfigParser.read()`, так как файл может отсутствовать, или может не быть разрешения на его открытие, что вызовет исключение.

## Отправка письма нескольким адресатам одновременно.

Для отправки письма нескольким адресатам одновременно необходимо вместо одного email-адреса, записанного в качестве отправителя, указать список адресатов. Чтобы это работало, нужно создать строку, разделенную запятой: `to_addr = "person-one@mail.ru, person-two@mail.ru"`, при создании переменной `body`, а также передать список адресов в `send_email()`.

```
import sys
import os
import configparser as ConfigParser

def send_email(to_addr, subject, text):
    # извлечение переменных из конфигурации
    server = cfg.get("smtp", "server")
    port = cfg.get("smtp", "port")
    from_addr = cfg.get("smtp", "email")
    passwd = cfg.get("smtp", "passwd")

    # формируем тело письма
    body = "\r\n".join((f"From: {from_addr}", f"To: {' '.join(to_addr)}",
                        f"Subject: {subject}", mime, charset, "", text))


    try:
        smtp = smtplib.SMTP(server, port)
        smtp.starttls()
        smtp.ehlo()
        smtp.login(from_addr, passwd)
        smtp.sendmail(from_addr, to_addr, body.encode(encode))
    except smtplib.SMTPException as err:
        print('Что - то пошло не так...')
        raise err
    finally:
        smtp.quit()

if __name__ == "__main__":
    # список адресов для отправки письма
    to_addr = ["person-one@mail.ru", "person-two@mail.ru"]
    subject = "Тестовое письмо от Python."
    text = "Отправкой почты управляет Python!"
    send_email(to_addr, subject, text)
```

# Использование строк TO, CC и BCC при отправке email.

В этом коде передадим три списка, в каждом из которых по одному адресу электронной почты. Создадим поля CC и BCC точно так же, как и раньше, а также необходимо объединить три списка адресов в один, чтобы передать его методу

РЕКЛАМА



new.sipuni.ru

ChatGPT повысит уровень ваших продаж!

Больше не придется тратить время на прослушку звонков и контроль менеджеров!

Цены

Регистрация

Оставить заявку

Заказать звонок

>

>

>

>

Узнать больше

```
import configparser
import smtplib
import ssl
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

def send_email(to_addr, subject, text, cc_eml, bcc_eml, encode='utf-8'):
    # извлечение переменных из конфигурации
    config = configparser.ConfigParser()
    config.read(os.path.abspath(__file__))
    (base_path, "email.ini")
    email_ini = os.path.join(base_path, "email.ini")
    config.read(email_ini)

    # получаем значения из конфигурации
    server = config.get("smtp", "server")
    port = config.get("smtp", "port")
    from_addr = config.get("smtp", "email")
    passwd = config.get("smtp", "passwd")
    charset = f'Content-Type: text/plain; charset={encode}'
    mime = 'MIME-Version: 1.0'

    # формируем тело письма
    body = "\r\n".join((f"From: {from_addr}", f"To: {' '.join(to_eml)}",
                        f"CC: {' '.join(cc_eml)}", f"BCC: {' '.join(bcc_eml)}",
                        f"Subject: {subject}", mime, charset, "", text))

    all_emails = to_eml + cc_eml + bcc_eml

    try:
        smtp = smtplib.SMTP(server, port)
        smtp.starttls()
        smtp.ehlo()
        smtp.login(from_addr, passwd)
        smtp.sendmail(from_addr, all_emails, body.encode(encode))
    except smtplib.SMTPException as err:
        print('Что - то пошло не так...')
        raise err
    finally:
        smtp.quit()

if __name__ == "__main__":
    to_addr = ["person@mail.ru"]
    cc_addr = ["person-one@mail.ru"]
    bcc_addr = ["person-two@mail.ru"]
    subject = "Тестовое письмо от Python."
    text = "Отправкой почты управляет Python!"
    send_email(to_addr, subject, text, cc_addr, bcc_addr)
```

# Добавление вложения/файла с помощью модуля email.

Но, [Вверх](#) email делает добавление вложений чрезвычайно простым.

https://docs-python.ru/tutorial/otpravka-email-vlozheniem/

5/7





```
attachment.set_payload(data)
encoders.encode_base64(attachment)
attachment.add_header(*header)
msg.attach(attachment)
```

РЕКЛАМА



new.sipuni.ru

ChatGPT повысит  
уровень ваших  
продаж!

Больше не придется тратить  
время на прослушку звонков  
и контроль менеджеров!

Цены

Регистрация

Оставить заявку

Заказать звонок

Узнать больше

```
...ытии файла вложения {file_to_attach}")

...rver, port)

...passwd)
...r, emails, msg.as_string())
...on as err:
...не так...')

..."]
...l.ru"]
...от Python."
...равляет Python!"
...le_attach'
...t, text, file_attach, cc_addr)
```

...ь электронные письма с помощью Python. Для тех, кто любит мини-проекты, нужно вернуть и добавить более подробную обработку ошибок в часть кода `smtp.sendmail()` на случай, если во время процесса произойдет что-то странное. Одним из примеров может быть добавление исключений `SMTPAuthenticationError` или `SMTPConnectError` и т. д.

Можно усилить обработку ошибок во время прикрепления файла, чтобы отловить другие ошибки. Наконец, можно взять различные списки адресов электронной почты и создать один нормализованный список, в котором будут удалены дубликаты. Это особенно важно, если читать список адресов из файла.

[DOCS-Python.ru™](#), 2023 г.

(Внимание! При копировании материала ссылка на источник обязательна)

[@docs\\_python\\_ru](#)

Вверх

<https://docs-python.ru/tutorial/otpravka-email-vlozheniem/>

7/7