



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 5

Дисциплина Компьютерные сети

Тема smtp-клиент

Вариант № 3

Студент Брянская Е.В.

Группа ИУ7-72Б

Оценка (баллы) \_\_\_\_\_

Преподаватель Рогозин Н.О.

Москва.  
2021 г.

## Задание

Написать smtp-клиент, который

- 1) В качестве входных данных (аргументы командной строки) получает:  
  
адрес получателя, адрес отправителя, пароль.
- 2) Использует **один из открытых** smtp-серверов для доставки MIME-сообщений, включая приложения, если они есть, в соответствии с вариантом.
- 3) Дополнительная задача зависит от варианта.

**I. Доставка сообщений выполняется с регулярным интервалом. Интервал и тело сообщения, имя файла для прикрепления (опционально) вводятся с клавиатуры.**

**II.** В качестве дополнительного параметра задается ключевое слово. По данному ключевому слову выполняется поиск в текстовых файлах в папке клиента. При обнаружении слова файл прикрепляется к письму.

**III.** Доставка сообщений выполняется в определенное время. Время и тело сообщения, имя файла для прикрепления (опционально) вводятся с клавиатуры.

Задача решается в несколько этапов.

### 1. Получение данных от пользователя

```
def input_data(data):
    print("--- Input data for sending mail ---")
    data["toUser"] = input("To: ")
    data["fromUser"] = input("From: ")
    data["password"] = input("Password: ")
    data["interval"] = input("Time interval: ")

    print("\n--- Mail ---")
    data["title"] = input("Title: ")
    data["body"] = input("Body: ")
    data["filename"] = input("Filename (optional): ")
```

### 2. Формирование письма

```
def create_msg(data):
    msg = MIMEMultipart()
    msg["From"] = data['fromUser']
    msg["To"] = data['toUser']
    msg["Date"] = formatdate(localtime=True)
    msg["Subject"] = data['title']
```

```

msg.attach(MIMEText(data['body']))

attachment = MIMEBase('application', "octet-stream")

if data["filename"] != '':
    try:
        with open(data["filename"], "rb") as f:
            tempData = f.read()

            attachment.set_payload(tempData)
            encoders.encode_base64(attachment)
            attachment.add_header('Content-Disposition', 'attachment;
filename="%s"' % data["filename"])

            msg.attach(attachment)
    except Exception:
        print("Error in open() file %s" % data["filename"])

return msg

```

### 3. Отправка

```

def send_mail(msg, data):
    smtp_ssl_host = 'smtp.yandex.ru'
    smtp_ssl_port = 465

    while True:
        try:
            server_ssl = smtplib.SMTP_SSL(smtp_ssl_host, smtp_ssl_port)
            server_ssl.login(data["fromUser"], data["password"])
            server_ssl.sendmail(data["fromUser"], data["toUser"],
msg.as_string())
            server_ssl.quit()

            print('Email was send')
        except Exception as exp:
            print("Error in sending mail %s" % exp)

        time.sleep(int(data["interval"]))

```

## Результаты

### Тест 1

--- Input data for sending mail ---

To: *bryanskayaelena@yandex.ru*  
From: *bryanskayakatya@yandex.ru*  
Password: *duxgvaukktoznmanz*  
Time interval: 5

--- Mail ---


Title: *Проверка*  
Body: *Успешно*  
Filename (optional):

BR	bryanskayakatya@yandex.ru	● Проверка	Успешно
BR	bryanskayakatya@yandex.ru	● Проверка	Успешно
BR	bryanskayakatya@yandex.ru	● Проверка	Успешно


## Тест 2

```
--- Input data for sending mail ---
To: bryanskayaelena@yandex.ru
From: bryanskayakatya@yandex.ru
Password: d\wXgvaukkoznmanz
Time interval: 5

--- Mail ---
Title: Проверка
Body: Успешно
Filename (optional): data\img.jpg
```

 bryanskayakatya@yandex.ru

Проверка  
Успешно



1

↓

## Полный код программы

```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email.utils import formatdate
from email import encoders

import smtplib
import time

def input_data(data):
    print("--- Input data for sending mail ---")
    data["toUser"] = input("To: ")
    data["fromUser"] = input("From: ")
    data["password"] = input("Password: ")
    data["interval"] = input("Time interval: ")

    print("\n--- Mail ---")
    data["title"] = input("Title: ")
    data["body"] = input("Body: ")
    data["filename"] = input("Filename (optional): ")

def create_msg(data):
    msg = MIMEMultipart()
    msg["From"] = data['fromUser']
    msg["To"] = data['toUser']
    msg["Date"] = formatdate(localtime=True)
    msg["Subject"] = data['title']
    msg.attach(MIMEText(data['body']))

    attachment = MIMEBase('application', "octet-stream")

    if data["filename"] != '':
        try:
            with open(data["filename"], "rb") as f:
                tempData = f.read()

            attachment.set_payload(tempData)
```

```

        encoders.encode_base64(attachment)
        attachment.add_header('Content-Disposition', 'attachment;
filename="%s"' % data["filename"])

    msg.attach(attachment)
    except Exception:
        print("Error in open() file %s" % data["filename"])
    return msg

def send_mail(msg, data):
    smtp_ssl_host = 'smtp.yandex.ru'
    smtp_ssl_port = 465

    while True:
        try:
            server_ssl = smtplib.SMTP_SSL(smtp_ssl_host, smtp_ssl_port)
            server_ssl.login(data["fromUser"], data["password"])
            server_ssl.sendmail(data["fromUser"], data["toUser"],
msg.as_string())
            server_ssl.quit()

            print('Email was send')
        except Exception as exp:
            print("Error in sending mail %s" % exp)

        time.sleep(int(data["interval"]))

if __name__ == "__main__":
    data = {}
    input_data(data)
    msg = create_msg(data)
    send_mail(msg, data)

```