Mandatory Assignment : SMTP Cleint

Submission Date and Time : check dtu learn

To be handed in a group on campusnet.

This assignment is obligatory.

The goal of this programming assignment is to create a mail client that sends e-mail to any recipient. Your client will need to establish a TCP connection with a mail server (e.g., use datacomm.bhsi.xyz on port 2526 to send mail and datacomm.bhsi.xyz on port 3001 to see if it was successful / or alternatively use datacommm.bhsi.xyz on port 25 and send mail from info@comit.dev to sXXXXXX@student.dtu.dk where XXXXXX is your student number), send relevant msgs as done through telnet with the mail server using the SMTP protocol, send an e-mail message to a recipient via the mail server, and finally close the TCP connection with the mail server. You can build on the exercise given on day 4 to write a client which can add two numbers and the smtp telnet exercises of day 3.

By the end of this assignment, you will have acquired a better understanding of SMTP protocol. You will also gain experience in implementing a standard protocol(socket programming) **using Python Java, C or any other language(you are not bound to any specific language)**

Your task is to develop a simple mail client that sends email to any recipient. Your client will need to connect to a mail server, dialogue with the mail server using the SMTP protocol, and send an email message to the mail server. Python provides a module, called smtplib, which has built in methods to send mail using SMTP protocol, similarly java also has similar libraries. However, you **should not** use these libraries in this assignment, because it hide the details of SMTP and socket programming.

In order to limit spam, some mail servers do not accept TCP connection from arbitrary sources. For the experiment described below, you may want to try connecting to my mail server to avoid complications(use telnet datacomm.bhsi.xyz 2526).

Assignment is based on the exercises from day 3 and day 4, to refresh please start by

1. Try the telnet exercise which showed you how the SMTP communication works.
2. Try the server client exercise from day 4 to understand the logic of how socket programming works.
3. Go through the slides from day 4 slides for socket programming (for python) and slides deck called socket programing in java from day 4 to see which methods, and objects you would typically need for such a program.
4. Read the RFC and learn about from ESMTP and extend your program to complete the extra credits task. You can also refer to file “email with attachment” in day5
5. Combine all aspects to complete the assignment

Code

The code provided below(for python programmers) and here for (Java Programmers) is only to help you get started. You are **not obligated to use this code**. Code and explanation- (remember this code is for reference and you can create program independent of this and the GUI, GUI is not mandatory)
https://media.pearsoncmg.com/aw/aw_kurose_network_3/labs/lab2/lab2.html

Remember GUI is not mandatory part of the assignment!!

Below you will find the skeleton code for the client. You are to complete the skeleton code. The places where you need to fill in code are marked with **#Fill in start** and **#Fill in end**. Each place may require one or more lines of code.

Additional Notes

In some cases, the receiving mail server might classify your e-mail as junk. Make sure you check the junk/spam folder when you look for the e-mail sent from your client.

In order to include images in your email, you would have to use the ESMTP which among other things also provides a way to send files(mandatory task 2). To read more details about ESMTP, are found here.

What to Hand in

In your submission, you are to provide the complete code for your SMTP mail client as well as a screenshot showing that you indeed receive the e-mail message from info@commit.dev assigned to you and send a reply to that mail. This means you from your mail client send a mail to yourself from the TA's dtu student mail address and then reply to it!!

Tasks

1. Write a SMTP mail client that can send mails.
2. [**extra credits**] Normally SMTP only handles sending text messages in the email body. Now, Modify your client such that it can send emails with both text and images. (hint: base64 can convert a image into a string)
3. [**extra credits**] Mail servers like Google mail (address: smtp.gmail.com, port: 587) requires your client to add a Transport Layer Security (TLS) or Secure Sockets Layer (SSL) for authentication and security reasons, before you send MAIL FROM command. Add TLS/SSL commands to your existing ones and implement your client using Google mail server and port. You can use the file called "telnet guide" from day 4 which shows how the TLS communication with google works in telnet.

Skeleton Python Code for the Mail Client

```
from socket import *


msg = "\r\n I love computer networks!"

endmsg = "\r\n.\r\n"


# Choose a mail server (e.g. Google mail server) and call it mailserver

mailserver = #Fill in start   #Fill in end


# Create socket called clientSocket and establish a TCP connection with mailserver

#Fill in start


#Fill in end

recv = clientSocket.recv(1024).decode()

print(recv)

if recv[:3] != '220':

       print('220 reply not received from server.')


# Send HELO command and print server response.

heloCommand = 'HELO Alice\r\n' # EHLO for extended SMTP

clientSocket.send(heloCommand.encode())

recv1 = clientSocket.recv(1024).decode()

print(recv1)

if recv1[:3] != '250':

    print('250 reply not received from server.')
```

```
# Send MAIL FROM command and print server response.

# Fill in start


# Fill in end


# Send RCPT TO command and print server response.

# Fill in start


# Fill in end


# Send DATA command and print server response.

# Fill in start


# Fill in end


# Send message data.

# Fill in start


# Fill in end

# Message ends with a single period.

# Fill in start


# Fill in end


# Send QUIT command and get server response.

# Fill in start
# Fill in end
```