Socket Programming Assignment 3: SMTP
Computer Networking

By:
Monil Shah
mds747

Python Code for the Mail Client:

1) Using free SMTP mail server not requiring SSL or TSL.

- Mail Server Used: mail.smtp2go.com
- Port Used: 2525

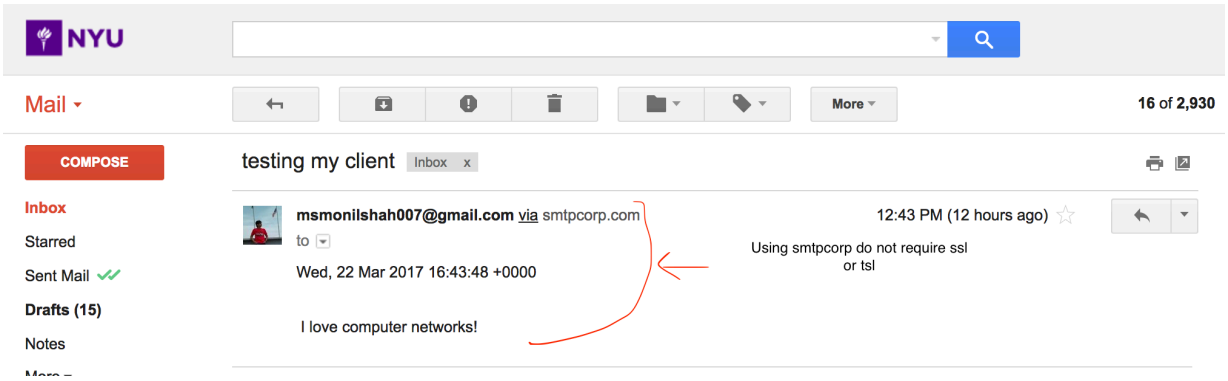
Python Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Mar 22 12:34:40 2017

@author: monilshah
"""



import socket
import base64
import time

msg = "\r\n I love computer networks!"
endmsg = "\r\n.\r\n"
mailserver = ('mail.smtp2go.com',2525)
#Fill in start
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientSocket.connect(mailserver)
```

```
#Fill in end
recv = clientSocket.recv(1024)
recv = recv.decode()
print("Message after connection request:" + recv)
if recv[:3] != '220':
    print('220 reply not received from server.')
# Send HELO command and print server response.
heloCommand = 'EHLO Alice\r\n'
clientSocket.send(heloCommand.encode())
recv1 = clientSocket.recv(1024)
recv1 = recv1.decode()
print("Message after EHLO command:" + recv1)
if recv1[:3] != '250':
    print('250 reply not received from server.')

#Info for username and password
username = "msmonilshah007@gmail.com"
password = “xxxxxx”
base64_str = ("\x00"+username+"\x00"+password).encode()
base64_str = base64.b64encode(base64_str)
authMsg = "AUTH PLAIN ".encode()+base64_str+"\r\n".encode()
clientSocket.send(authMsg)
recv_auth = clientSocket.recv(1024)
print(recv_auth.decode())
# Send MAIL FROM command and print server response.
# Fill in start
mailFrom = "MAIL FROM:msmonilshah007@gmail.com\r\n"
clientSocket.send(mailFrom.encode())
recv2 = clientSocket.recv(1024)
recv2 = recv2.decode()
print("After MAIL FROM command: "+recv2)
# Fill in end
# Send RCPT TO command and print server response.
# Fill in start
rcptTo = "RCPT TO:mds747@nyu.edu\r\n"
clientSocket.send(rcptTo.encode())
```

```
recv3 = clientSocket.recv(1024)
recv3 = recv3.decode()
print("After RCPT TO command: "+recv3)
# Fill in end
# Send DATA command and print server response.
# Fill in start
data = "DATA\r\n"
clientSocket.send(data.encode())
recv4 = clientSocket.recv(1024)
recv4 = recv4.decode()
print("After DATA command: "+recv4)
# Fill in end
# Send message data.
# Fill in start
subject = "Subject: testing my client\r\n\r\n"
clientSocket.send(subject.encode())
date = time.strftime("%a, %d %b %Y %H:%M:%S +0000", time.gmtime())
date = date + "\r\n\r\n"
clientSocket.send(date.encode())
clientSocket.send(msg.encode())
clientSocket.send(endmsg.encode())
recv_msg = clientSocket.recv(1024)
print("Response after sending message body:"+recv_msg.decode())
# Fill in end
# Send QUIT command and get server response.
# Fill in start
quit = "QUIT\r\n"
clientSocket.send(quit.encode())
recv5 = clientSocket.recv(1024)
print(recv5.decode())
# Fill in end
clientSocket.close()
```

ScreenShot of Email received:



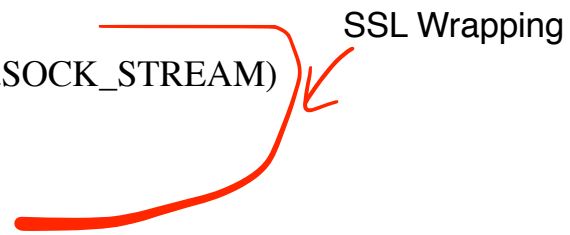
2) Extra Credit : Using Google mail server and SSL

- Mail Server Used: smtp.gmail.com
- Port Used: 465

Python Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 23 01:42:31 2017

@author: monilshah
"""

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Mar 22 12:34:40 2017

@author: monilshah
"""
```

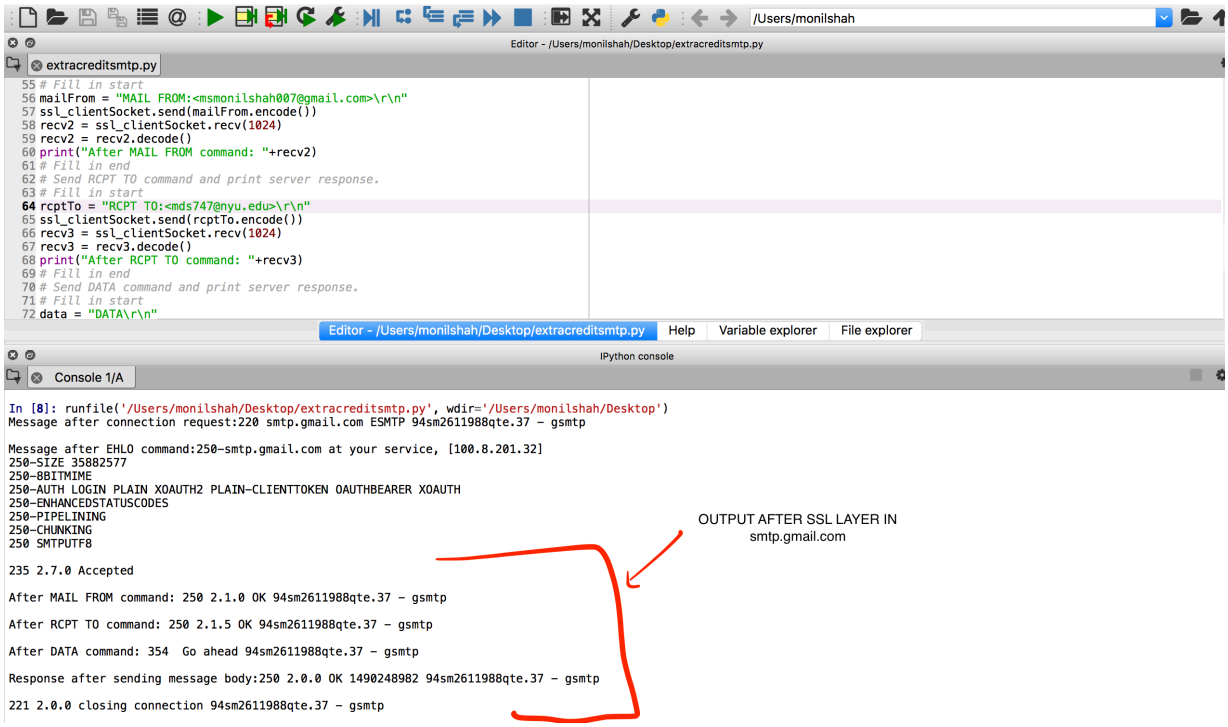
```
import socket
import base64
import time
import ssl
msg = "\r\n I love computer networks!"
endmsg = "\r\n.\r\n"
mailserver = ('smtp.gmail.com',465)
#Fill in start
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
ssl_clientSocket = ssl.wrap_socket(clientSocket)
ssl_clientSocket.connect(mailserver)
#Fill in end
recv = ssl_clientSocket.recv(1024)
recv = recv.decode()
print("Message after connection request:" + recv)
if recv[:3] != '220':
    print('220 reply not received from server.')
# Send HELO command and print server response.
heloCommand = 'EHLO Alice\r\n'
ssl_clientSocket.send(heloCommand.encode())
recv1 = ssl_clientSocket.recv(1024)
recv1 = recv1.decode()
print("Message after EHLO command:" + recv1)
if recv1[:3] != '250':
    print('250 reply not received from server.')

#Info for username and password
username = "msmonilshah007@gmail.com"
password = “xxxxxxxx”
base64_str = ("\x00"+username+"\x00"+password).encode()
base64_str = base64.b64encode(base64_str)
authMsg = "AUTH PLAIN ".encode()+base64_str+"\r\n".encode()
ssl_clientSocket.send(authMsg)
recv_auth = ssl_clientSocket.recv(1024)
print(recv_auth.decode())
```



```
# Send MAIL FROM command and print server response.
# Fill in start
mailFrom = "MAIL FROM:<msmonilshah007@gmail.com>\r\n"
ssl_clientSocket.send(mailFrom.encode())
recv2 = ssl_clientSocket.recv(1024)
recv2 = recv2.decode()
print("After MAIL FROM command: "+recv2)
# Fill in end
# Send RCPT TO command and print server response.
# Fill in start
rcptTo = "RCPT TO:<mds747@nyu.edu>\r\n"
ssl_clientSocket.send(rcptTo.encode())
recv3 = ssl_clientSocket.recv(1024)
recv3 = recv3.decode()
print("After RCPT TO command: "+recv3)
# Fill in end
# Send DATA command and print server response.
# Fill in start
data = "DATA\r\n"
ssl_clientSocket.send(data.encode())
recv4 = ssl_clientSocket.recv(1024)
recv4 = recv4.decode()
print("After DATA command: "+recv4)
# Fill in end
# Send message data.
# Fill in start
subject = "Subject: testing my client\r\n\r\n"
ssl_clientSocket.send(subject.encode())
date = time.strftime("%a, %d %b %Y %H:%M:%S +0000", time.gmtime())
date = date + "\r\n\r\n"
ssl_clientSocket.send(date.encode())
ssl_clientSocket.send(msg.encode())
ssl_clientSocket.send(endmsg.encode())
recv_msg = ssl_clientSocket.recv(1024)
print("Response after sending message body:"+recv_msg.decode())
# Fill in end
```

```
# Send QUIT command and get server response.
# Fill in start
quit = "QUIT\r\n"
ssl_clientSocket.send(quit.encode())
recv5 = ssl_clientSocket.recv(1024)
print(recv5.decode())
# Fill in end
ssl_clientSocket.close()
```

ScreenShot of Program Running Successfully:



ScreenShot of Email received: