

SMTP client using gmail servers

Sharad Shivmath

N13619376

Note:

- password modified due to privacy/security reasons
- please modify recipient, sender, username, password to verify

code:

```
__author__ = 'sharads'

import socket
import base64

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

recipient = "<sss617@nyu.edu>"
sender = "<sharad.shivmath@gmail.com>"
username = "sharad.shivmath"
password = 'password' # password string modified

# Choose a mail server (e.g. Google mail server) and call it mailserver
mailserver = "smtp.gmail.com"
port = 587

# Create socket called clientSocket and establish a TCP connection with
mailserver
clientSocket = socket.socket()
clientSocket.connect((mailserver, port))
recv = clientSocket.recv(1024)
print recv
if recv[:3] != '220':
    print '220 reply not received from server.'

# Send HELO command and print server response.
clientSocket.send('HELO Alice\r\n')
recv1 = clientSocket.recv(1024)
print recv1
if recv1[:3] != '250':
    print '250 reply not received from server.'
```

```

# Request an encrypted connection
clientSocket.send('STARTTLS\r\n')
tlsRecv = clientSocket.recv(1024)
print tlsRecv
if tlsRecv[:3] != '220':
    print '220 reply not received from server'

# Encrypt the socket
ssl_clientSocket = socket.ssl(clientSocket)

# Send the AUTH LOGIN command and print server response.
ssl_clientSocket.write('AUTH LOGIN\r\n')
authRecv = ssl_clientSocket.read(1024)
print authRecv
if authRecv[:3] != '334':
    print '334 reply not received from server'

# Send username and print server response.
uname = base64.b64encode(username) + '\r\n'
ssl_clientSocket.write(uname)
unameRecv = ssl_clientSocket.read(1024)
print unameRecv
if unameRecv[:3] != '334':
    print '334 reply not received from server'

# Send password and print server response.
pword = base64.b64encode(password) + '\r\n'
ssl_clientSocket.write(pword)
pwordRecv = ssl_clientSocket.read(1024)
print pwordRecv
if pwordRecv[:3] != '235':
    print '235 reply not received from server'

# Send MAIL FROM command and print server response.
mailFromCommand = 'MAIL FROM: ' + sender + '\r\n'
ssl_clientSocket.write(mailFromCommand)
recv2 = ssl_clientSocket.read(1024)
print recv2
if recv2[:3] != '250':
    print '250 reply not received from server.'

# Send RCPT TO command and print server response.
rcptToCommand = 'RCPT TO: ' + recipient + '\r\n'

```

```
ssl_clientSocket.write(rcptToCommand)
recv3 = ssl_clientSocket.read(1024)
print recv3
if recv3[:3] != '250':
    print '250 reply not received from server.'

# Send DATA command and print server response.
dataCommand = 'DATA\r\n'
ssl_clientSocket.write(dataCommand)
recv4 = ssl_clientSocket.read(1024)
print recv4
if recv4[:3] != '354':
    print '354 reply not received from server.'

# Send message data.
ssl_clientSocket.write(msg)

# Message ends with a single period.
ssl_clientSocket.write(endmsg)
recv5 = ssl_clientSocket.read(1024)
print recv5
if recv5[:3] != '250':
    print '250 reply not received from server.'

# Send QUIT command and get server response.
ssl_clientSocket.write('QUIT\r\n')
recv6 = ssl_clientSocket.read(1024)
print recv6
if recv6[:3] != '221':
    print '221 reply not received from server.'

clientSocket.close()
```

