

Socket Programming 2-Email Client

Source Code:

```
1  from socket import *
2
3  # Create a variable to hold the message and end message
4  msg = "\r\n I love computer networks!"
5  endmsg = "\r\n.\r\n"
6
7  # Choose a mail server (e.g. Google mail server) and call it mailserver
8  mailserver = 'smtp.csus.edu'
9  mailport = 25
10
11 # Create socket called clientSocket and establish a TCP connection with mailserver
12 clientSocket = socket(AF_INET, SOCK_STREAM)
13 clientSocket.connect((mailserver,mailport))
14
15 # Recieve the response from the server
16 recv = clientSocket.recv(1024).decode()
17
18 # Print the recieved message
19 print(recv)
20
21 # Check to see if the recieved message is not equal to 220
22 if recv[:3] != '220':
23
24     # If it is something other than 220 then we output that we did not recieve a 220 reply
25     print('220 reply not received from server.')
26
27
28 # Create a variable to hold the HELO command
29 helloCommand = 'HELO Alice\r\n'
30
31 # Create a client socket and send the encoded HELO command
32 clientSocket.send(helloCommand.encode())
33
34 # Recieve the response from the server
35 recv = clientSocket.recv(1024).decode()
36
37 # Print the recieved message
38 print(recv)
39
40 # Check to see if the recieved message is not equal to 250
41 if recv[:3] != '250':
42
43     # If it is something other than 250 then we output that we did not recieve a 250 reply
44     print('250 reply not received from server.')
45
46
```

```

47 # Create a variable to hold the MAIL FROM command
48 mailFromCommand = 'MAIL FROM: igoroleshko1@gmail.com\r\n'
49
50 # Create a client socket and send the encoded MAIL FROM command
51 clientSocket.send(mailFromCommand.encode())
52
53 # Recieve the respose from the server
54 recv = clientSocket.recv(1024).decode()
55
56 # Print the recieved message
57 print(recv)
58
59 # Check to see if the recieved message is not equal to 250
60 if recv[:3] != '250':
61
62     # If it is something other than 250 then we output that we did not recieve a 250 reply
63     print('250 reply not received from server.')
64
65 # Create a variable to hold the RCPT TO command
66 receiptToCommand = 'RCPT TO: ioleshko@csus.edu\r\n'
67
68 # Create a client socket and send the encoded RCPT TO command
69 clientSocket.send(receiptToCommand.encode())
70
71 # Recieve the respose from the server
72 recv = clientSocket.recv(1024).decode()
73
74 # Print the recieved message
75 print(recv)
76
77 # Check to see if the recieved message is not equal to 250
78 if recv[:3] != '250':
79
80     # If it is something other than 250 then we output that we did not recieve a 250 reply
81     print('250 reply not received from server.')
82
83 # Create a variable to hold the DATA command
84 dataCommand = 'DATA\r\n'
85
86 # Create a client socket and send the encoded DATA command
87 clientSocket.send(dataCommand.encode())
88
89 # Recieve the respose from the server
90 recv = clientSocket.recv(1024).decode()
91
92 # Print the recieved message
93 print(recv)
94
95 # Check to see if the recieved message is not equal to 354
96 if recv[:3] != '354':
97
98     # If it is something other than 354 then we output that we did not recieve a 354 reply
99     print('354 reply not received from server.')
100

```

```

101 # Create a client socket and send the encoded msg data
102 clientSocket.send(msg.encode())
103
104 # Create a client socket and send the encoded endmsg data
105 clientSocket.send(endmsg.encode())
106
107 # Recieve the respose from the server
108 recv = clientSocket.recv(1024).decode()
109
110 # Print the recieved message
111 print(recv)
112
113 # Check to see if the recieved message is not equal to 250
114 if recv[:3] != '250':
115
116     # If it is something other than 250 then we output that we did not recieve a 250 reply
117     print('250 reply not received from server.')
118
119 # Create a variable to hold the QUIT command
120 quitCommand = 'QUIT\r\n'
121
122 # Create a client socket and send the encoded QUIT command
123 clientSocket.send(quitCommand.encode())
124
125 # Recieve the respose from the server
126 recv = clientSocket.recv(1024).decode()
127
128 # Print the recieved message
129 print(recv)
130
131 # Check to see if the recieved message is not equal to 221
132 if recv[:3] != '221':
133
134     # If it is something other than 221 then we output that we did not recieve a 221 reply
135     print('221 reply not received from server.')
136

```

SMTP Interaction:

```
>>> %Run smtp.py
220 smtp.saclink.csus.edu Microsoft ESMTP MAIL Service ready at Wed, 5 Apr 2023 17:41:20 -0700

250 smtp.saclink.csus.edu Hello [10.117.215.138]

250 2.1.0 Sender OK



250 2.1.5 Recipient OK



354 Start mail input; end with <CRLF>.<CRLF>


250 2.6.0 <32d63d8b-53d5-41ca-a495-5b473b318539@irt-pa-e16mbx01.saclink.csus.edu> [InternalId=35
278861370137, Hostname=irt-pa-e16mbx02.saclink.csus.edu] 1620 bytes in 0.130, 12.158 KB/sec Queu
ed mail for delivery


221 2.0.0 Service closing transmission channel
```







Email Creation

(No subject)  





igoroleshko1@gmail.com 




Wed 4/5/2023 5:58 PM

I love computer networks!

 Reply

 Reply all

 Forward