How can I have multiple clients on a TCP Python Chat Server?

6

3

Any help on how I can get this to accept more than one client, and why it isn't at the moment? Thanks!

Also, is there anything I'm doing wrong with this code? I've been following mostly Python 2 tutorials because I can't find any for Python 3.4

Here is my Server code:

```
import socket
import time
import os
from threading import Thread

folderPath = "Chat Logs"
filePath = folderPath + "/" + str(time.strftime("%H-%M-%S_%d-%m-%Y")) + ".txt"

def clientHandler(c):
    while True:
        data = c.recv(1024)
        if not data:
            break

    data = data.decode("UTF-8")

    message = str(data[:data.index("§")])
    nick = str(data[data.index("§")+1:])

        print(nick + ": " + message)
        saveChat(nick, message)
        print("   Sending: " + data)
        c.send(bytes(data, "UTF-8"))

    c.close()

def saveChat(nick, message):
    if not os.path.exists(folderPath):
        os.makedirs(folderPath)
    if not os.path.exists(filePath):
        f = open(filePath, "a")
        f.close()

    f = open(filePath, "a")
    f.write(nick + ": " + message + "\n")
    f.close()

def Main():
    host = str(socket.gethostbyname(socket.gethostname()))
    port = 5000

    print(host + ":" + str(port) + "\n")
    Clients = int(input("Clients: "))

    s = socket.socket()
```

```
            print("Connection from: " + str(addr))

            Thread(target=clientHandler(c)).start()
        s.close()

    if __name__ == "__main__":
        Main()
```

And here is my Client code:

```
import socket

def Main():
    print("Send 'q' to exit\n")
    address = str(input("ip:port -> "))
    nick = input("nick: ")

    try:
        if address.index(":") != 0:
            host = address[:address.index(":")]
            port = int(address[address.index(":")+1:])
    except ValueError:
        host = address
        port = 5000

    s = socket.socket()
    s.connect((host, port))

    message = input("-> ")

    while message != "q":
        s.send(bytes(message + "□□" + nick, "UTF-8"))
        data = s.recv(1024)
        data = data.decode("UTF-8")
        data2 = data

        messageServer = str(data[:data.index("□□")])
        nickServer = str(data[data.index("□□")+1:])
        if not data == data2:
            print(nickServer + ": " + messageServer)
        message = input("-> ")
    s.close()

if __name__ == "__main__":
    Main()
```

python tcp python-multithreading

edited Oct 18 '14 at 23:02

asked Oct 18 '14 at 22:49
artman41
150 1 1 12

10

First of all, I found these tutorials very helpful: BinaryTides

Here is an example of a simple tcp server that accepts multiple clients. All this one does receive data from the client and return "OK .. " + the_data. However, you could easily modify it to have a function that broadcasts the data(chat msg) to all clients connected. This example uses threading. You should google for the `select` module. With regards to your threads, are you sure you are a) using the right module/method for the job and b) that you are calling it in the right way?

```
import socket
import sys
from thread import start_new_thread

HOST = '' # all availabe interfaces
PORT = 9999 # arbitrary non privileged port

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
except socket.error, msg:
    print("Could not create socket. Error Code: ", str(msg[0]), "Error: ",
msg[1])
    sys.exit(0)

print("[-] Socket Created")

# bind socket
try:
    s.bind((HOST, PORT))
    print("[-] Socket Bound to port " + str(PORT))
except socket.error, msg:
    print("Bind Failed. Error Code: {} Error: {}".format(str(msg[0]), msg[1]))
    sys.exit()

s.listen(10)
print("Listening...")

# The code below is what you're looking for ############

def client_thread(conn):
    conn.send("Welcome to the Server. Type messages and press enter to send.\n")

    while True:
        data = conn.recv(1024)
        if not data:
            break
        reply = "OK . . " + data
        conn.sendall(reply)
    conn.close()

while True:
    # blocking call, waits to accept a connection
    conn, addr = s.accept()
    print("[-] Connected to " + addr[0] + ":" + str(addr[1]))

    start_new_thread(client_thread, (conn,))

s.close()
```

I get the error `ImportError: No module named 'thread'` when trying to do the import for `start_new_thread` – artman41 Oct 19 '14 at 14:02

Ah I see, look here, this may shed some light on that. stackoverflow.com/questions/5568555/thread-vs-threading – Totem Oct 19 '14 at 17:46

ah, thanks for the info @Totem – artman41 Oct 21 '14 at 10:10

1 This code gives the error: [Errno 10048] Only one usage of each socket address (protocol/network address/port) is normally permitted – Matt Jul 28 '17 at 14:39

3

Check out: http://etutorials.org/Programming/Python+tutorial/Part+IV+Network+and+Web+Programming/Chapter+19.+Sockets+and+Server-Side+Network+Protocol+Modules/19.3+Event-Driven+Socket+Programs/ . Example **19-6** is (the one with the `select` system call) like a hello world of chat applications. You might also want to take a look at http://beej.us/guide/bgnet/output/html/multipage/index.html for more lower level (C) insight system networking basics.

answered Oct 18 '14 at 23:05

PSkocik
38.5k 6 59 84

Thank you for the help, going to check them now :) – artman41 Oct 18 '14 at 23:05