

# 北京邮电大学

Socket Programming Assignment : Web Server



academy : 信息与通信工程学院  
class : 2015211107  
number : 2015210187  
Order : 11  
name : 方森彪  
data : 2018 年 12 月 8 日

# 1,request

In this lab, you will learn the basics of socket programming for TCP connections in Python: how to create a socket, bind it to a specific address and port, as well as send and receive a HTTP packet. You will also learn some basics of HTTP header format. You will develop a web server that handles one HTTP request at a time. Your web server should accept and parse the HTTP request, get the requested file from the server's file system, create an HTTP response message consisting of the requested file preceded by header lines, and then send the response directly to the client. If the requested file is not present in the server, the server should send an HTTP "404 Not Found" message back to the client.

## 2.Code

### (1)Python code

```
#import socket module

from socket import *

serverSocket = socket(AF_INET, SOCK_STREAM)

#Prepare a sever socket

serverSocket.bind(('', 6789)) # 将 TCP 欢迎套接字绑定到
指定端口

serverSocket.listen(1) # 最大连接数为 1

while True:

    #Establish the connection

    print('Ready to serve...')

    connectionSocket, addr = serverSocket.accept() # 接
收到客户连接请求后, 建立新的 TCP 连接套接字

    try:

        message = connectionSocket.recv(1024) # 获取客户
发送的报文

        filename = message.split()[1]

        f = open(filename[1:])

        outputdata = f.read();

        #Send one HTTP header line into socket
```

```

        header = ' HTTP/1.1 200 OK\nConnection:
close\nContent-Type: text/html\nContent-
Length: %d\n\n' % (len(outputdata))

        connectionSocket.send(header.encode())

        #Send the content of the requested file to the
client

        for i in range(0, len(outputdata)):
            connectionSocket.send(outputdata[i].encode())
        connectionSocket.close()
    except IOError:
        #Send response message for file not found

        header = ' HTTP/1.1 404 Not Found'

        connectionSocket.send(header.encode())

        #Close client socket

        connectionSocket.close()
serverSocket.close()

```

## (2)html content

```

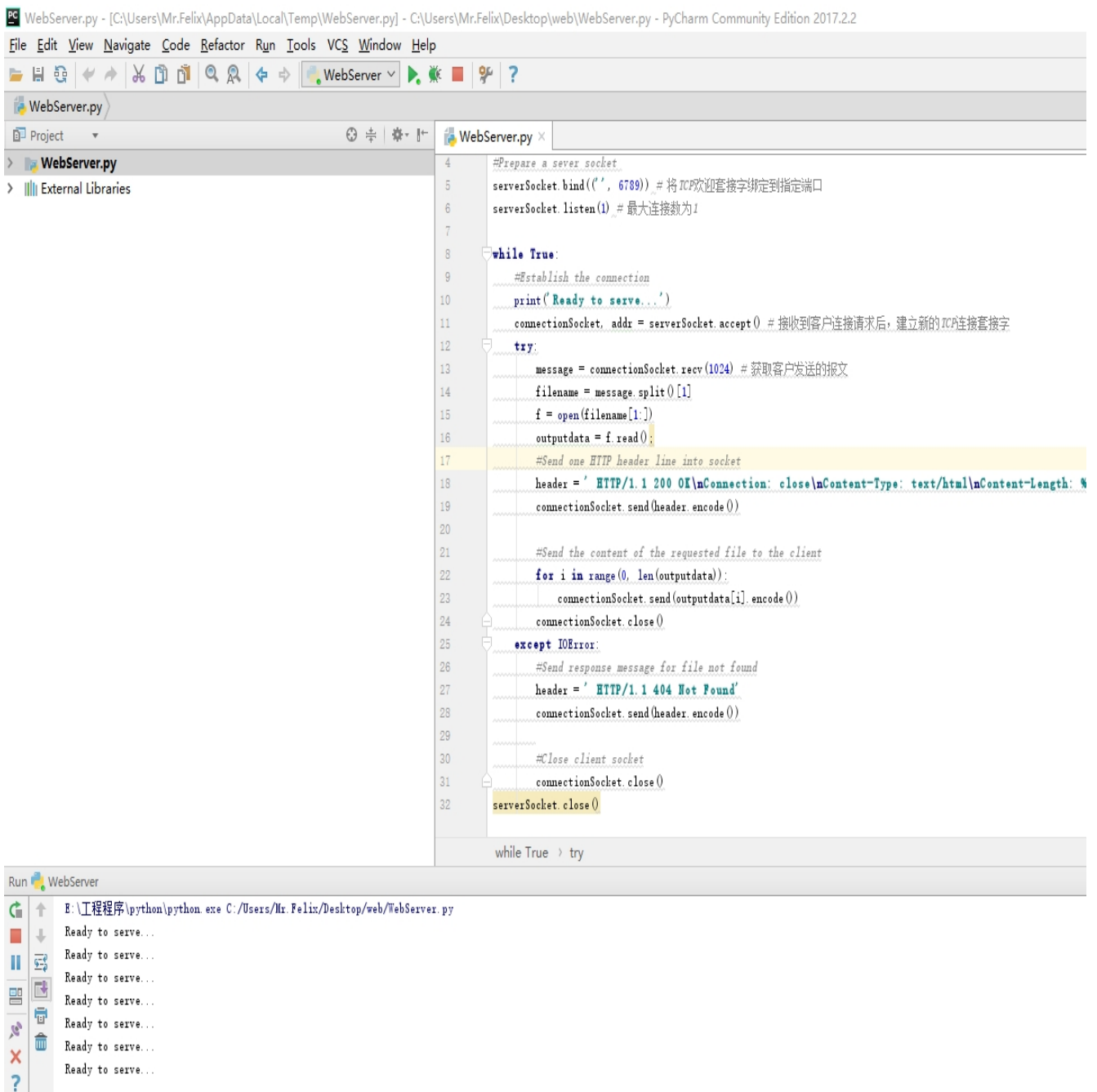
<head>FANG SENBIAO  Felix.
<br>What an amazing website! </head>

```

### 3,screen shot

I use my iphone to connect the web server, and the following is the screen shot of the process.

start of the web server



The screenshot displays the PyCharm IDE interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The main editor window shows the code for `WebServer.py`. The code is a Python script that sets up a simple web server using sockets. It binds to port 6789 and listens for connections. When a connection is accepted, it prints 'Ready to serve...', receives a message, splits it to get the filename, opens the file, reads its content, and sends an HTTP response header and the file content back to the client. It also handles the case where a file is not found by sending a 404 response. The server socket is closed at the end of the loop.

```
4 #Prepare a sever socket
5 serverSocket.bind(('', 6789)) # 将TCP欢迎套接字绑定到指定端口
6 serverSocket.listen(1) # 最大连接数为1
7
8 while True:
9     #Establish the connection
10    print('Ready to serve...')
11    connectionSocket, addr = serverSocket.accept() # 接收到客户连接请求后, 建立新的TCP连接套接字
12    try:
13        message = connectionSocket.recv(1024) # 获取客户发送的报文
14        filename = message.split()[1]
15        f = open(filename[1:])
16        outputdata = f.read()
17        #Send one HTTP header line into socket
18        header = ' HTTP/1.1 200 OK\r\nConnection: close\r\nContent-Type: text/html\r\nContent-Length: %s' % len(outputdata)
19        connectionSocket.send(header.encode())
20
21        #Send the content of the requested file to the client
22        for i in range(0, len(outputdata)):
23            connectionSocket.send(outputdata[i].encode())
24        connectionSocket.close()
25    except IOError:
26        #Send response message for file not found
27        header = ' HTTP/1.1 404 Not Found'
28        connectionSocket.send(header.encode())
29
30        #Close client socket
31        connectionSocket.close()
32    serverSocket.close()
33
34 while True > try
```

The bottom panel shows the Run console output, which displays the command used to run the script and the output 'Ready to serve...' repeated multiple times, indicating the server is running and ready to accept connections.