Assignment 1

Abdelrahman Mahmoud Kamal Mahmoud Nour 37

Problem Statement:

Using sockets to implement a simple web client that communicates with a web server using a restricted subset of HTTP. The main objective of this assignment is to give you hands-on experience with UNIX sockets.

Data Structures:

- STL library data structures such as:

  Vectors

  Unordered_maps

- And C++ basic data structures like char*, strings ..etc

- HTTP class:

Overall organization of the server:

The server consists of the following ..

- Main function:

  Accepts the connection and delegates it to a thread which executes the handle_connection function to parse the request and send the required data.

- Connection_handler:

  It parses the request and sends the required data in the form of http request.

  It also determine the method of the request and sends the suitable responses for each method

- Http_parser:

  It parses the http request sent to extract its body and information.

- File_handler:

  Which handles and creates directories if not found.

Major Functions of the server:

- DWORD WINAPI handle_conncetion(LPVOID lpParameter)
- void parse_http(char *message ,

  unordered_map<string,string> &headers ,

  string &method ,

  string &filename ,

  string &http_version ,

  string &body ,

  int size_of_message)

And other utility functions.

Overall organization of the client:

The server consists of the following ..

- Main function:

  It calls the file parser to extract commands and send it to the server and receives the messages and displays it.

- File_parser:

  It parses the input file to extract commands.

- Http_parser:

  It parses the http request sent to extract its body and information.

- File_handler:

  Which handles and creates directories if not found.

The server timeouts after a period of time using select() functions if there are no requests for that time on the same connection.

Major Functions of the client:

- parse_input_file("input.txt");
- void parse_http(char *message ,

  unordered_map<string,string> &headers ,

  string &method ,

  string &filename ,

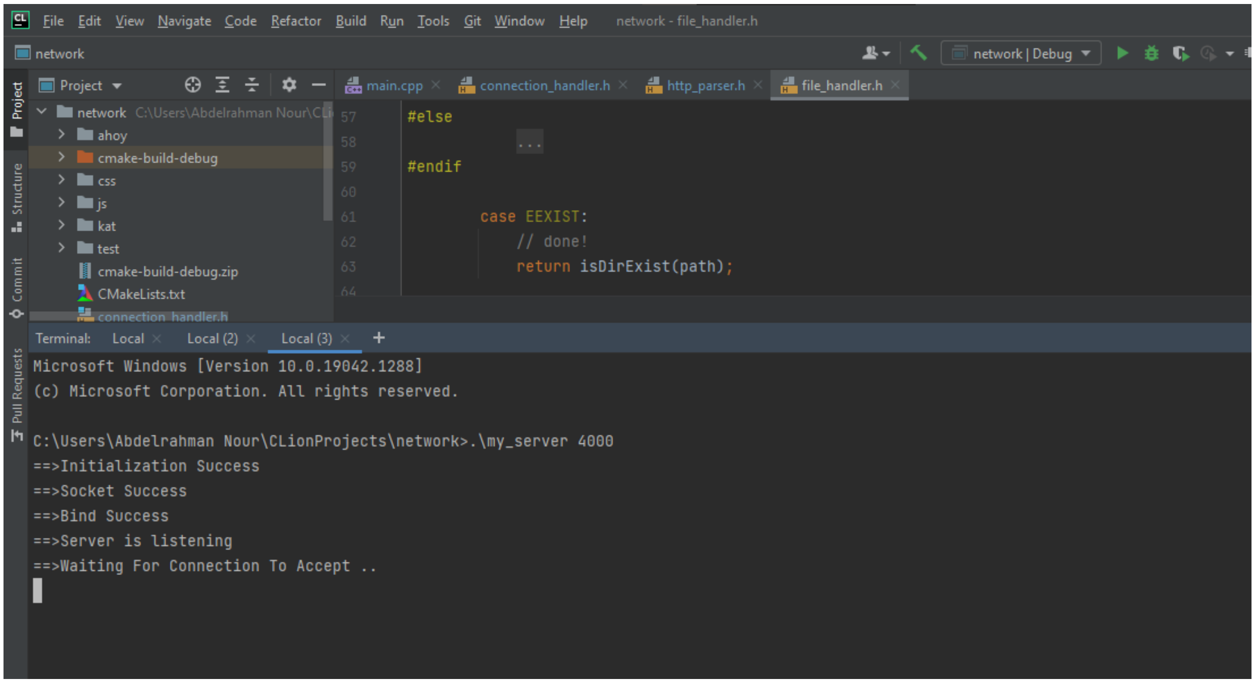
  string &http_version ,

  string &body ,

  int size_of_message)

And other utility functions.

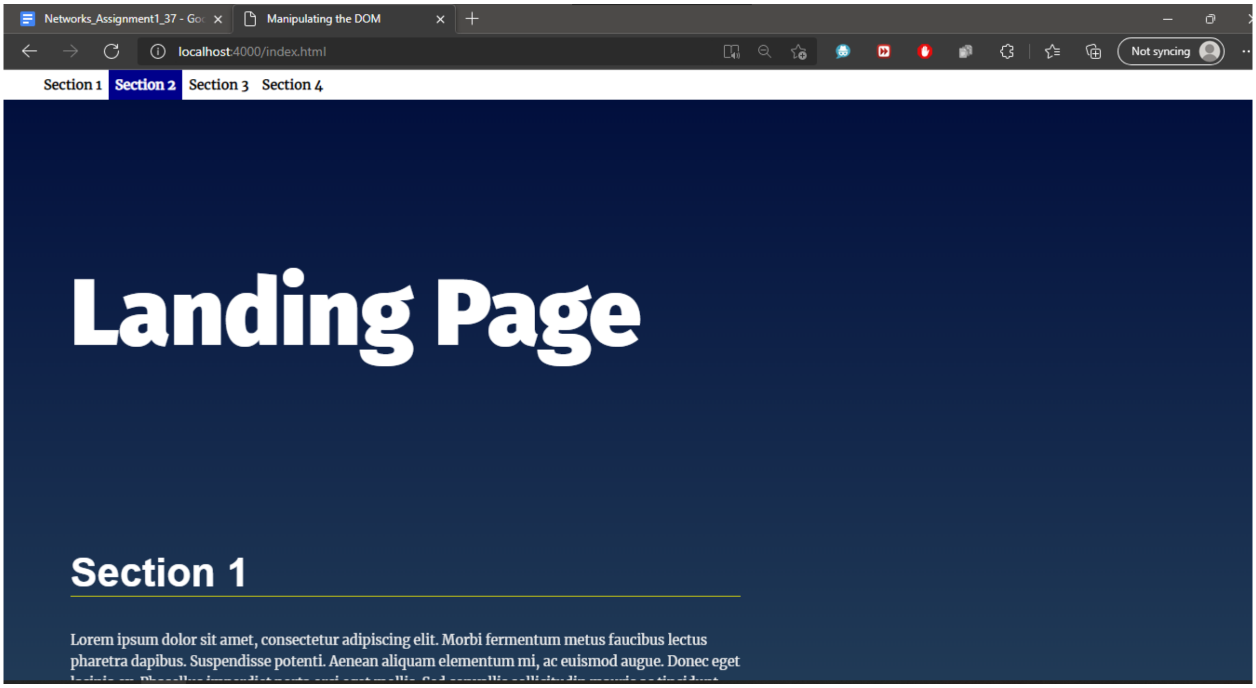
Test the server with browser:



```
#else
        ...
#endif

        case EEXIST:
            // done!
            return isDirExist(path);
```

```
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Abdelrahman Nour\CLionProjects\network>.\my_server 4000
==>Initialization Success
==>Socket Success
==>Bind Success
==>Server is listening
==>Waiting For Connection To Accept ..
```

Running the server

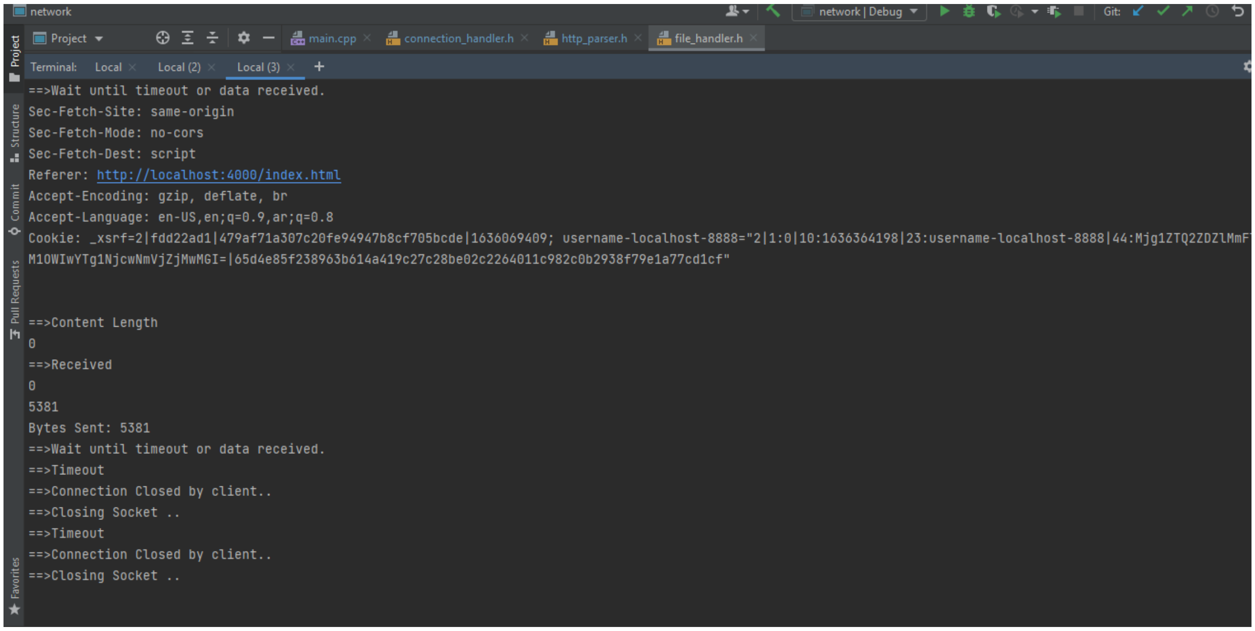


Landing Page

Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi fermentum metus faucibus lectus pharetra dapibus. Suspendisse potenti. Aenean aliquam elementum mi, ac euismod augue. Donec eget

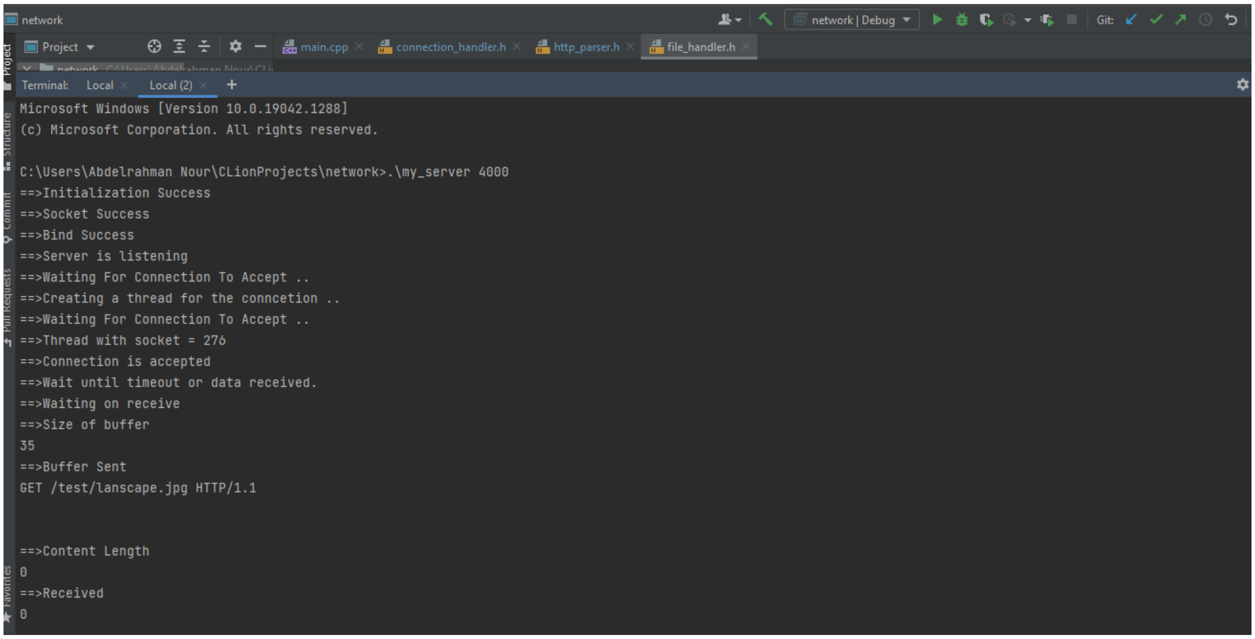
Testing using the browser



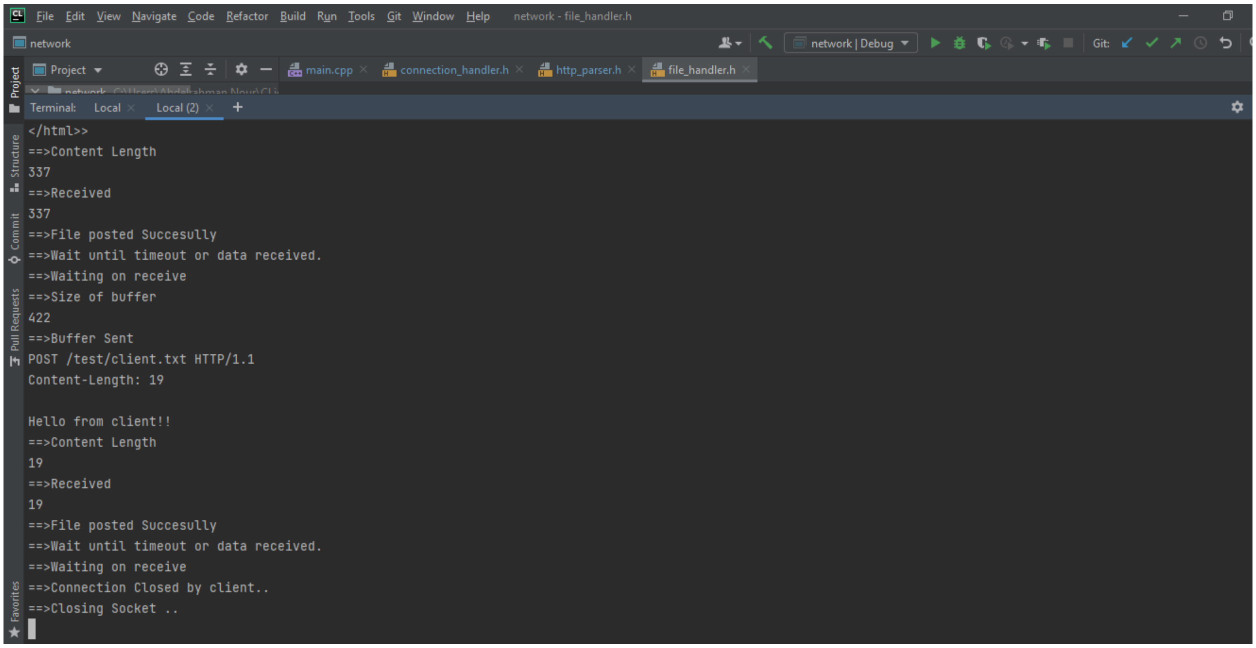
```
==>Wait until timeout or data received.
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: script
Referer: http://localhost:4000/index.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ar;q=0.8
Cookie: _xsrf=2|fdd22ad1|479af71a307c20fe94947b8cf705bcde|1636069409; username-localhost-8888="2|1:0|10:1636364198|23:username-localhost-8888|44:Mjg1ZTQ2ZDZlMmF
M1OWIwYTg1NjcwNmVjZjMwMGI=|65d4e85f238963b614a419c27c28be02c2264011c982c0b2938f79e1a77cd1cf"


==>Content Length
0
==>Received
0
5381
Bytes Sent: 5381
==>Wait until timeout or data received.
==>Timeout
==>Connection Closed by client..
==>Closing Socket ..
==>Timeout
==>Connection Closed by client..
==>Closing Socket ..
```

Client and server:

Server

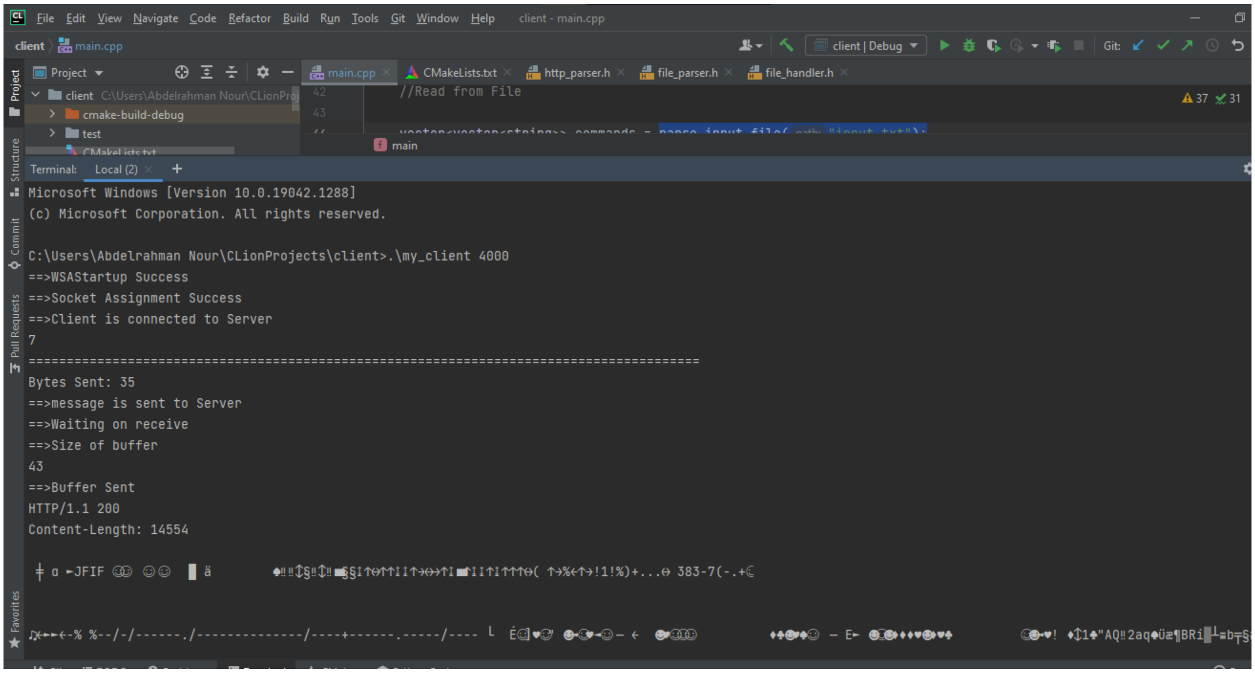


```
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Abdelrahman Nour\CLionProjects\network>.\my_server 4000
==>Initialization Success
==>Socket Success
==>Bind Success
==>Server is listening
==>Waiting For Connection To Accept ..
==>Creating a thread for the conncetion ..
==>Waiting For Connection To Accept ..
==>Thread with socket = 276
==>Connection is accepted
==>Wait until timeout or data received.
==>Waiting on receive
==>Size of buffer
35
==>Buffer Sent
GET /test/lanscape.jpg HTTP/1.1


==>Content Length
0
==>Received
0
```



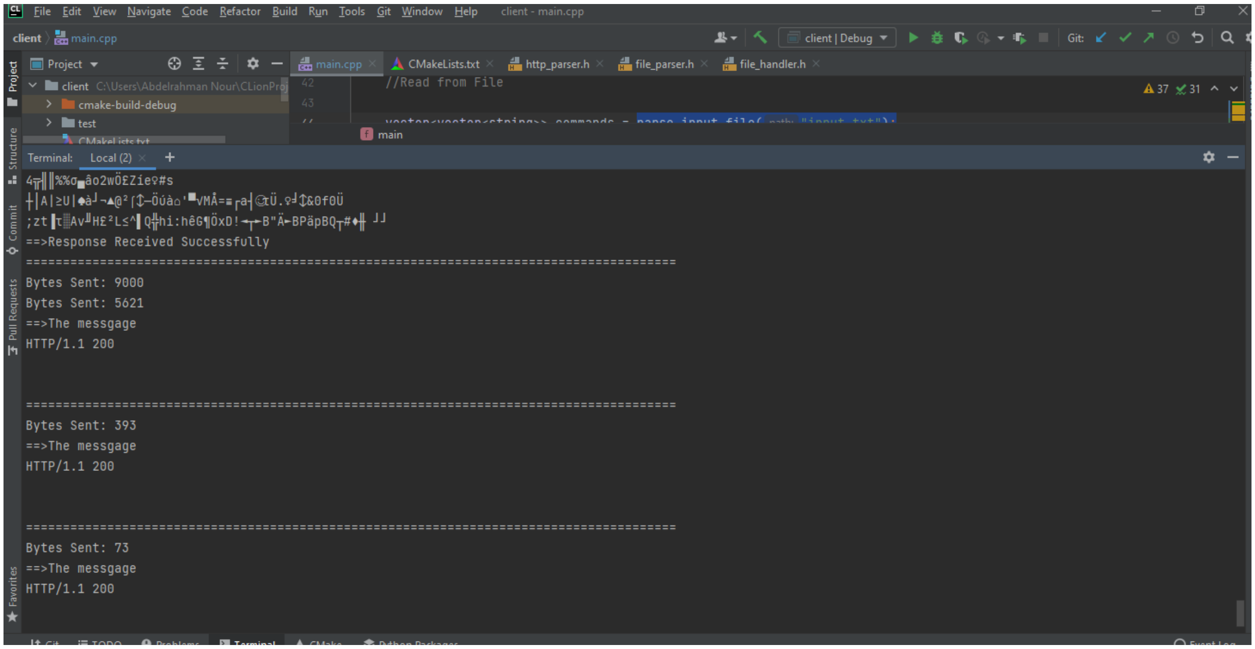
```
</html>>
==>Content Length
337
==>Received
337
==>File posted Succesully
==>Wait until timeout or data received.
==>Waiting on receive
==>Size of buffer
422
==>Buffer Sent
POST /test/client.txt HTTP/1.1
Content-Length: 19

Hello from client!!
==>Content Length
19
==>Received
19
==>File posted Succesully
==>Wait until timeout or data received.
==>Waiting on receive
==>Connection Closed by client..
==>Closing Socket ..
```

Client:



```
42        //Read from File
43
44        vector<vector<string>> commands = parse_input_file( path: "input.txt");
```

```
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Abdelrahman Nour\CLionProjects\client>.\my_client 4000
==>WSAStartup Success
==>Socket Assignment Success
==>Client is connected to Server
7
=========================================================================================
Bytes Sent: 35
==>message is sent to Server
==>Waiting on receive
==>Size of buffer
43
==>Buffer Sent
HTTP/1.1 200
Content-Length: 14554

 ╪ α ►JFIF ☺☺  ☺☺  ▐ ä        ♠‼‼↕§‼↕‼▄§§↨↑↔↑↑↨↨↑→↔→↑↨▄↑↨↨↑↨↑↑↑↔( ↑→%←↑→!1!%)+...↔ 383-7(-.+∈


♫←►►←-% %--/-/------./-------------/----+------.-----/---- └  É☺♥☺  ☻☺♥-☺— ←  ☻♥☺☺☺          ♦♣☻♥♠☺  — E►  ☻☺☻♦♦♦♥☻♦♥♣         ☺☻♥! ♦↕1♠"AQ‼2aq♠Üæ¶BRí▒┴≡b╤§
```



```
42        //Read from File
43
44        vector<vector<string>> commands = parse_input_file( path: "input.txt");
```

```
4╥║║%%σ▄âo2wÖ£Zíe♀#s
┼|A|≥U|♠à┘¬▲@²⌠↕─Öúà⌂'▀vMÅ=≡┌a┤☺τÜ.♀┘↕&0f0Ü
;zt▐τ▒Av╜H£²L≤^▌Q╬hi:hêG¶ÖxD!◄┬►B"Ä►BPäpBQ┬#♦╫ ┘┘
==>Response Received Successfully
=========================================================================================
Bytes Sent: 9000
Bytes Sent: 5621
==>The messgage
HTTP/1.1 200


=========================================================================================
Bytes Sent: 393
==>The messgage
HTTP/1.1 200


=========================================================================================
Bytes Sent: 73
==>The messgage
HTTP/1.1 200
```