

Socket Programming Assignment: Web Server

For this assignment, use basic socket programming for TCP connections in Python. You will

- create a socket
- bind it to a specific address and port
- send and receive HTTP messages
- use the HTTP header formats as we did with telnet

Task

Develop a web server that handles one HTTP request at a time. Your web server will accept and parse an HTTP request message, get the requested file from the server's file system (you must create this file as well, see below), create an HTTP response message consisting of the requested file preceded by appropriate message header lines, and send the response directly to the client.

If the requested file is not present in the server, your server will send an HTTP "404 Not Found" message back to the client.

Running the Server

- 1) Put a simple HTML file (*e.g.*, HelloWorld.html) in the same directory that your web server is in.
 - a. For examples of *simple* .html pages, you can use google. One good, simple example is at (and this has *much* more than you need to include) http://help.websiteos.com/websiteos/example_of_a_simple_html_page.htm
 - b. You only need to create something with a very simple message that can be loaded by your web browser.
 - c. This is a two week assignment to allow you extra time for this investigation of using HTML to create a webpage.
- 2) Run the server program that you have written.
- 3) Determine the IP address of your host that is running the server. From the same or a different host, open a browser and provide the corresponding URL (IP+port+file).
 - a. For example: <http://131.229.102.177:1400/hello.html>
 - b. 'hello.html' is the name of the file you placed in your server directory.

Note the use of the port number after the colon. You need to replace this port number with whatever port you have used in your server code. The browser should then display the contents of hello.html.
 - c. What happens if you forget the port number? Why?
- 4) Next try to get a file that is not present at the server. You should get the "404 Not Found" message that your server sends to your web browser (client).

What to Hand in

Feb 15: Hand in your HTML code that your web server will include in its response message, and a first draft (presumably not working yet) of your Python code for the server.

Feb 22: Hand in the complete, well commented, server code along with the screen shots of your client browser, verifying that you actually receive the contents of the HTML file from the server.

Notes for Idle

To navigate around the file (directory) tree, use the os library. Helpful commands are `os.getcwd()` and `os.chdir()`