

# Csci 4131 Internet Programming

Lecture 14, March 11<sup>th</sup>  
Spring 2024

**Instructor: Dr. Dan Challou**

# Logistics (Csci 4131, Lecture 14, March 11th)

- Zybooks HW 7 due Sunday 3/17 (***topics are key to doing HW 5 successfully !!!***)
- Homework 4 due next Friday 3/22
- Homework 5 will be out next week – Using Fetch or AJAX, JSON, Node.js

# Readings/Tutorials: Node.js, JSON, Fetch, AJAX – For HW 5!

## **Node.js References and Tutorials:**

Your zyBook

<https://www.w3schools.com/nodejs/>

<https://codeburst.io/the-only-nodejs-introduction-youll-ever-need-d969a47ef219>

## **JSON References / Tutorials:**

Your zyBook

[https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)

[https://www.w3schools.com/js/js\\_json.asp](https://www.w3schools.com/js/js_json.asp)

[www.json.org](http://www.json.org)

Optional: Chapter 10.3.3 Sebesta

## **FETCH References / Tutorials:**

Your Zybook

[https://www.w3schools.com/js/js\\_api\\_fetch.asp](https://www.w3schools.com/js/js_api_fetch.asp)

<https://javascript.info/fetch>

## **AJAX References / Tutorials:**

Your Zybook

[https://www.w3schools.com/xml/ajax\\_intro.asp](https://www.w3schools.com/xml/ajax_intro.asp)

Optional: Sebesta, Chapter 10 © Dan Challou, 2024, All Rights Reserved.

Do not share or reproduce without the  
express written consent of the author

# Questions ?

# Agenda

- HW 4
- JavaScript Object Notation (JSON)
- AJAX / Fetch

# HW4 Demo Video Available

- In Media Gallery on Class Canvas Site!

# HW-4 (and beyond) related items

# Programs you can use to send HTTPS requests to servers (so you can test YOUR server)

> telnet

telnet> help

... telnet help menu ...

telnet> open 127.0.0.1 8000 (*127.0.0.1 = localhost*)

Connecting To 127.0.0.1...

GET /index.html HTTP/1.0

(Hit enter twice to send the terminating blank line ...)

... HTTP response message ...

**Telnet is a character-based protocol – *so no typos when entering a command (can't backspace)***



# OR Can use a Network Program

```
import java.net.*;
import java.io.*;

public class HttpClient {
    public static void main(String[] args) throws IOException {
        // The host and port to be connected.
        String host = "127.0.0.1";
        int port = 8000;
        // Create a TCP socket and connect to the host:port.
        Socket socket = new Socket(host, port);
        // Create the input and output streams for the network socket.
        BufferedReader in
            = new BufferedReader(
                new InputStreamReader(socket.getInputStream()));
        PrintWriter out
            = new PrintWriter(socket.getOutputStream(), true);
        // Send request to the HTTP server.
        out.println("GET /index.html HTTP/1.0");
        out.println(); // blank line separating header & body
        out.flush();
        // Read the response and display on console.
        String line;
        // readLine() returns null if server close the network socket.
        while((line = in.readLine()) != null) {
            System.out.println(line);
        }
        // Close the I/O streams.
        in.close();
        out.close();
    }
}
```

# Or, can use

- Unix/Linux **cURL** command, which supports HTTP scripting
  - <https://terminalcheatsheet.com/guides/curl-rest-api>
  - <https://www.baeldung.com/curl-rest>
- Or, write a python3 program that uses the **requests** module
  - <https://www.geeksforgeeks.org/create-api-tester-using-python-requests-module/>
- Or Unix / Linux **wget** command
  - <https://techs-tricks.blogspot.com/2008/12/test-http-request-with-wget.html>
- Or you can use POSTMAN application on your own machines.
  - <https://www.geeksforgeeks.org/testing-rest-api-with-postman-and-curl/>
- BUT, you must test your stuff on a machine where you can set file permissions to ensure your server will return a 403 error when files with no access permissions are requested
  - e.g. `—rw-r----- aFile.html`

# Additional References for applications you can use to test your server

- The Unix/Linux **cURL** command, which supports HTTP scripting
  - [https://www.tutorialspoint.com/unix\\_commands/curl.htm](https://www.tutorialspoint.com/unix_commands/curl.htm)
  - We have also posted a pdf named: **CurlTesting.pdf** with the lecture 7 materials
- Or the Unix / Linux **wget** command
  - [http://www.tutorialspoint.com/unix\\_commands/wget.htm](http://www.tutorialspoint.com/unix_commands/wget.htm)
- Or, the PostMan utility:
  - <https://www.getpostman.com/postman>
- Python 3 requests:
  - [https://www.w3schools.com/python/module\\_requests.asp](https://www.w3schools.com/python/module_requests.asp)

# And, of course

- The address bar of your browser
- And the **MyServer.html** page provided with the assignment
- To test Post requests, you can also use the form from your HW2 or HW3

# Yet event more HW 4 Notes

Python modules you may need (and you can use) - *in addition to the modules used in the code we have provided for HW4*

**os**

**os.path.join** – to create a path to file in the current directory (recall that dot “.” is the current directory, and the URI will be the resource you are looking for - specified in the first line of the HTTP request message)

**os.path.exists** – tells you if a resource exists

**os.stat** – to get the permissions on the resource, which are stored in the **st\_mode** attribute

ALTERNATIVELY – see

**os.access**

see- [https://www.tutorialspoint.com/python/os\\_access.htm](https://www.tutorialspoint.com/python/os_access.htm)

**stat**

**sys**

And,

**from urllib.parse import unquote**

the unquote utility to unquote the **values** sent in the *name&value* pairs in a post request

# Toward Building The Back-End of our Full – Stack Web Site

- For HW5, and HW6 We'll use **node.js** to build the WebServer for the back end of our website
- Why?
  - Node.js is in JavaScript, and there are lots of nice free frameworks available for it (we will use express). It is more widely used than Python or Python Frameworks for building web servers. And finally, we are getting comfortable JavaScript and one of the course objectives is to learn JavaScript

# Technologies needed for HW 5 and beyond

- All the stuff from the first ½ of the course plus:
- Node.js (You will use it to construct a HTTP server)
- JSON
- FETCH
  - OR
- AJAX

# Recall Node.js :

info obtained from:

[https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp)

[https://www.w3schools.com/nodejs/nodejs\\_get\\_started.asp](https://www.w3schools.com/nodejs/nodejs_get_started.asp)

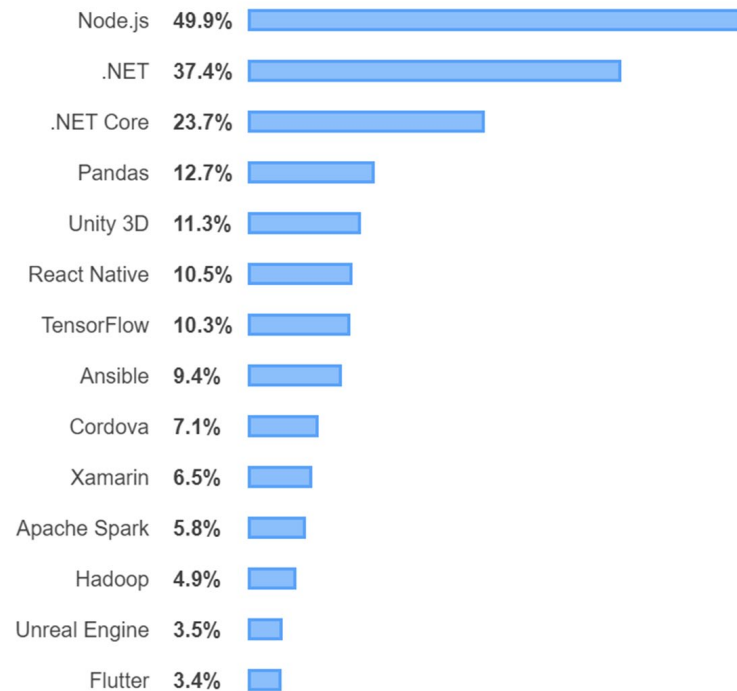
[https://www.w3schools.com/nodejs/nodejs\\_modules.asp](https://www.w3schools.com/nodejs/nodejs_modules.asp)

- Node.js is an open source **server** framework
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript to implement and augment **server** functionality



# Why Node.js ?

According to [Stack Overflow 2019 survey](#), Node.js had become one of the most popular tools in “Frameworks, Libraries, and Tools” category with 50% of answers!



# Node.js Standings in Stack Overflow developers Survey 2021 – Programming, Scripting and Markup languages



<https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-web-frameworks>

© Dan Challou, 2024, All Rights Reserved.

Do not share or reproduce without the  
express written consent of the author

# Another Source, Since I often have my doubts about Stack Overflow

- <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

# A Node.js file contains:

- tasks that will be executed when triggered by certain events
  - A typical event is someone trying to access a port on the server
  - Node.js files must be initiated on the server before having any effect
- Node.js files have extension ".js"

# Like Python, Node.js has lots of libraries (called modules) that you will want to include in your application

- For example:
  - http module, used to create a server

# You can run node.js on your personal computer

- <https://nodejs.org/en/download>

# Next Time

- Node.js revisited
- JSON Revisited
- Introduction to Fetch, AJAX?