

Csci 4131

JSON Wrap-up

Node.js intro

AJAX Revisited

Lecture 16, October 29th

Fall 2018

Dr. Dan Challou

Logistics

- Exam 1 is this coming Wednesday 10/31; In this classroom (Anderson 370) during the class time (2:30 – 3:45pm). Open Book, Open notes. No Electronics (Computers, Phones, Smart Watches, Google Googles, Alexa, ..., etc.)
- HW 5 – `node.js/AJAX/json` with a calendar page and a Form is out and due Friday, November 9

Logistics, continued

- **An Important Note on Exam I:**

If you don't notify me **BEFORE** the exam that you can't make it **AND** you don't have a university sanctioned excuse **AND** you are not here for the exam, you get a zero on the Exam

Note, you can notify me and all TA's of any issue in this regard via the Class Email:

csci4131f18_help@umn.edu

Slides Describing Exam Scope, Overview will be posted with Lecture 15 Materials (October 24th)

- File name:

Csci4131Exam1_Info.pdf

Reading and Tutorials: JSON, Ajax, Node.js

- JSON
 - Sebesta – Chapters 10, Section 3.3
 - https://www.w3schools.com/js/js_json_intro.asp
 - <https://www.json.org/>
- AJAX
 - Sebesta – Chapter 10
 - https://www.w3schools.com/xml/ajax_intro.asp
- Node.js
 - <https://www.w3schools.com/nodejs/default.asp>
 - <https://www.tutorialspoint.com/nodejs/>
 - <https://nodejs.org/en/docs/guides>

Node.js training videos on Lynda

- You can access Lynda via the following link:

<http://lynda.umn.edu>

Use your x.500 id and password to sign in.

The following videos are most helpful:

- 1. Node.js Essential Training (6h 22m - Detailed Node.js video)
- 2. Building a Website with Node.js and Express.js (3h 16m - Focusses on Express.js and Node.js)
- 3. Learning Node.js (1h 57m)

Last Time

- HW 4 wrap up
- Review HTTP error messaging exercise
- JSON wrap up
- Intro to AJAX

Questions?

Today

- JSON Wrap-up
- Intro to Node.js
- AJAX revisited

Review Exercise from Last Class

1. Create an HTML page with a **div** element. The div element should have an id named: **locations**
 2. Add the JavaScript necessary to do the following:
 3. Store the following TEXT in a JavaScript Variable in a JSON format:
 4. "lat1": "44.95045", "lon1": "-93.345002"
 5. "lat2": "44.95045", "lon2": "-93.345002"
 6. Convert it to the text to a JSON object using JSON parse
 7. Next, write JavaScript necessary to display the latitudes (**lat**) and longitudes (**lon**) in a list on the div element with the id named: **locations**
- Example: [jsonexer1.html](#)

(Hint – look at the json examples from last lecture)

© Dan Chailou, 2018. All Rights Reserved.

Do not copy or redistribute without the
express written consent of the Author.

Node.js Revisited

(info obtained from:

https://www.w3schools.com/nodejs/nodejs_intro.asp
https://www.w3schools.com/nodejs/nodejs_get_started.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

Node.js can:

- generate dynamic page content
- create, open, read, write, delete, and close files on the server
- can collect form data
- can add, delete, modify data in your database

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"

Node.js handles a file request as follows

1. Sends the task to the computer's file system.
 2. Ready to handle the next request.
 3. When the file system has opened and read the file, the server returns the content to the client.
- Thus node.js is single threaded, non-blocking, and asynchronous
 - Here is how Php handles a file request:
 1. Sends the task to the computer's file system.
 2. Waits while the file system opens and reads the file.
 3. Returns the content to the client.
 4. Ready to handle the next request.
 - So, for this task, PHP (and ASP) operate synchronously (and block).

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

Example of a node.js file

```
var http = require('http');  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.end('Hello World!');  
}).listen(8080);
```

Assuming it is in the file: myfirst.js

You run it from the command line by typing:

node myfirst.js

Then fire up your browser, and in the address bar type:

http://localhost:8080

And, you will get the response: Hello World – rendered in your browser

HW 5 Overview, Discussion

AJAX revisited

Recall AJAX From a few classes back

- AJAX is defined as Asynchronous Javascript and JSON
- JSON is stored in a text file!!!!
- JSON is obtained as text, then parsed into a JavaScript object using **JSON.parse** (the parse method on the global JavaScript JSON object)
- To turn a JSON object back into a string use **JSON.stringify(obj)**

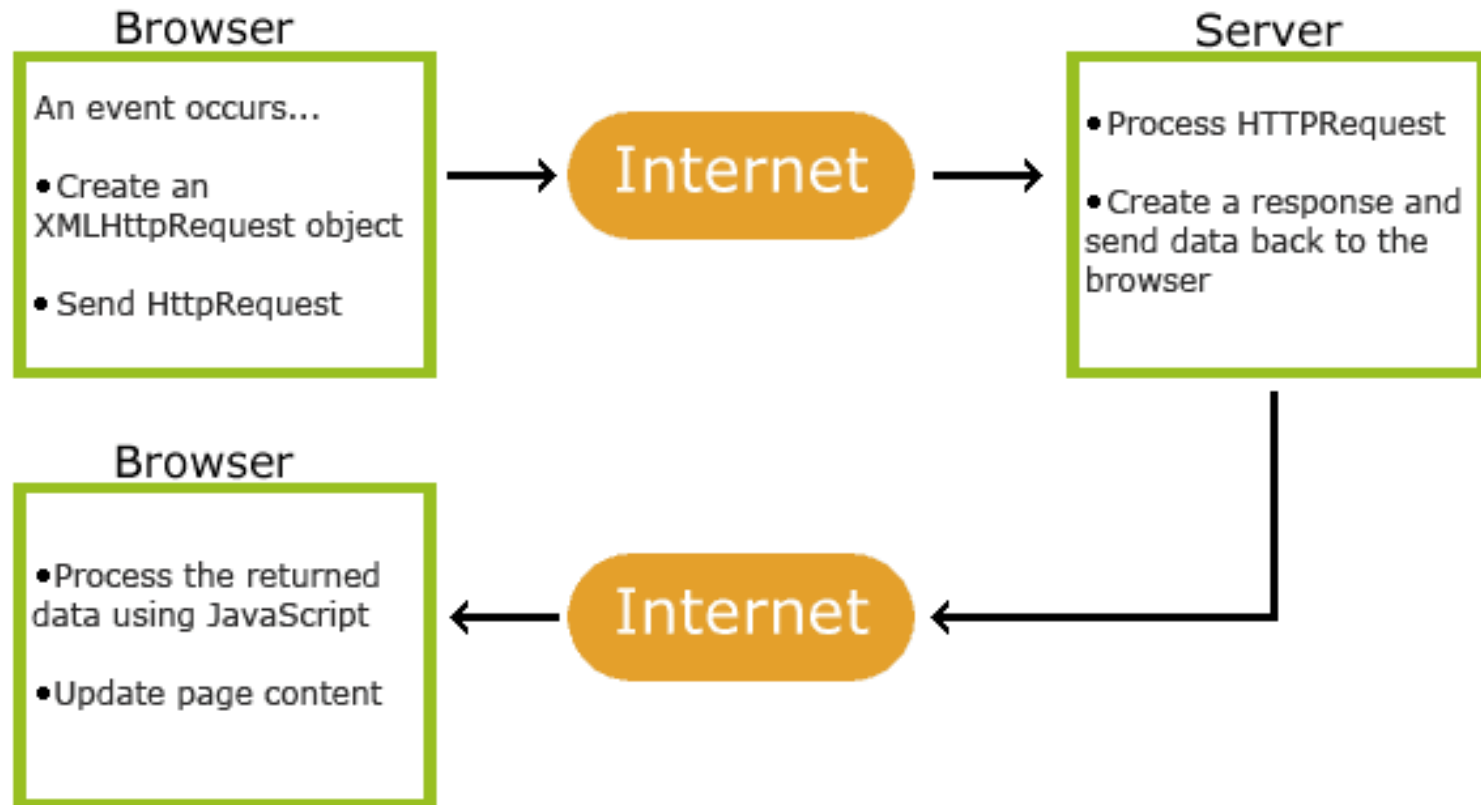
Recall example of Reading a JSON file using JavaScript and XMLHttpRequest()

- <http://www-users.cs.umn.edu/~challou/JSON/JSONHttpRequest.html>

The name AJAX (or AJAJ) is a bit of a misnomer

- Asynchronous JavaScript can be used to retrieve data stored in various formats including:
 - Text
 - Images
 - JSON
 - XML
 - ???

How Does AJAX Work (After Getting HTML, CSS, JAVASCRIPT FILES FROM SERVER)?



Source: http://www.w3schools.com/php/php_ajax_intro.asp

The XMLHttpRequest Object

- This is the backbone of AJAX
- The XMLHttpRequest object is used to exchange data with a server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Creating an XMLHttpRequest Object

- Syntax for creating an XMLHttpRequest object:
variable=new XMLHttpRequest();

Key Event for : The **onreadystatechange** Event

- When a request to a server is sent, we want to perform some actions based on the response.
- The onreadystatechange event is triggered every time the readyState changes.
- The readyState property holds the status of the XMLHttpRequest.
- In the onreadystatechange event, we specify what will happen when the server response is ready to be processed.

Source:http://www.w3schools.com/ajax/ajax_xmlhttprequest_onreadystatechange.asp

Three Important Properties of the onreadystatechange event:

When status == 200, and state =4, we have obtained the response from our initial request

Property	Description
onreadystatechange	Stores a function (or the name of a function) to be called automatically each time the readyState property changes
readyState	Holds the status of the XMLHttpRequest. Changes from 0 to 4: 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 404: Page not found

©Dan Challou, 2018. All Rights Reserved.

Do not copy or redistribute without the express written consent of the Author.

AJAX Adheres to A “Same Origin” Policy

- https://en.wikipedia.org/wiki/Same-origin_policy

Another Example of AJAX in Action

- Example

<http://www-users.cs.umn.edu/~challou/simpleAJAXex.html>

Next Time

- Exam 1 – Here in Anderson 370 from 2:30 to 3:45pm