

Csci 4131

AJAX Revisited

Lecture 18, Nov 5th

Fall 2018

Dr. Dan Challou

Logistics

- HW 5 – node.js/AJAX/json with a calendar page and a Form is out and is due:

Friday, November 9 at 2pm.

Late Submission (with penalty), Nov 10 at
2am

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>
 - <https://nodejs.org/api/index.html>

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

- Exam 1

Exam 1 Results

Q1	Q2	Q3	Q4	Q5	Q6	Avg/ Std Dev
16.1	17.8	16.2	12.96	9.5	11.9	84.42
2.94	1.72	2.75	2.95	1.19	3.24	9.65

Exam 1 Procedure for Perceived Issues

- We will not start accepting requests for regrades until the end of Class on Wednesday 11/7
- Review your test
- Run your answer
- If you think you have an issue:
 - **Do not change or write anything on your test.** We have scanned them. If you change something, request a regrade, and we note it, you have cheated and will get a ZERO on the exam
 - On a single piece of paper , write down your issue(s)
 - Feelings are good in general, but not objective and thus not considered when regrading
 - We will only grade based on our interpretation of what you have written or drawn on your test.
- Hand in the paper along with your test to any TA or Dr. Dan

Exam 1 Information Continued

- The window for Issues requests opens Wednesday 11/7 after class and closes Wednesday 11/14 after class

Questions?

Today

- AJAX revisited
- HW 5 Revisited

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)**

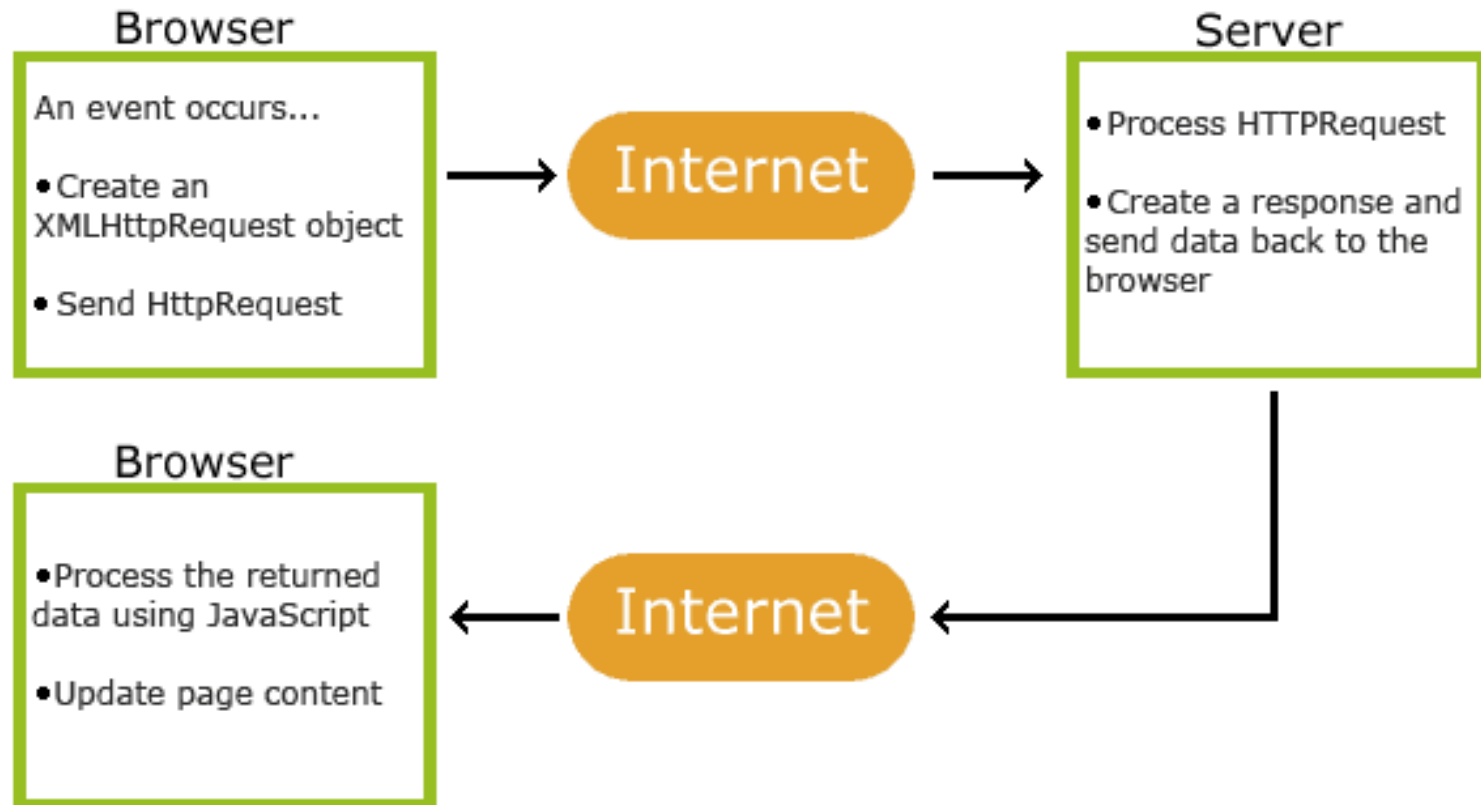
Recall example of Reading a JSON file using JavaScript and XMLHttpRequest()

- <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
<code>onreadystatechange</code>	Stores a function (or the name of a function) to be called automatically each time the <code>readyState</code> property changes
<code>readyState</code>	Holds the status of the <code>XMLHttpRequest</code> . 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
<code>status</code>	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.

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

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>

HW 5 Overview, Discussion

Next Time

- HW 5 Revisited Again (We will address the AddCalendar page and adding an item to the form)
- Exam 1 Review