Csci 4131
Intro to JSON

Lecture 14, October 22nd

Fall 2018

Dr. Dan Challou

Logistics

- HW 4 is due this coming Friday, October 26th at 2:00pm. After that time, assignments will be accepted with penalty though EARLY Saturday Morning (2:00am), 10/27

  After 2:00am 10/27 – HW 4 submissions

  will not be accepted.
- Exam 1 is Wednesday 10/31; In this classroom at the usual class time. Open Book, Open notes. No Electronics

Reading - HTTP Protocol

- Foundation of the WWW
- Target – impart a deeper understanding of how it works, along with a better understanding of browser and web server function
- **Reading:**
  - **http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html**
  - RFC 2616 (HTTP 1.1)

    https://tools.ietf.org/html/rfc2616

    http://www.w3c.org/Protocols/

    https://www.jmarshall.com/easy/http/ (This is a nice site too)

And, Starting 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

Last Time

- Reviewed HTTP Client Server Architecture
- Application Level Protocol in ISO 7-layer network architecture
- HTTP Request and Response Messages
- HTTP Methods
- HTTP Errors
- Refactored Presentation Level Server (EchoServer, on class Moodle Site and reviewed in c) to respond to HTTP REQUEST Message (HEAD Method) and Posted the Code to the Class Moodle Site (in a pdf file)

Reminder besides your Browser, utilities you can use to test HW4

- The Unix/Linux **cURL** command, which supports HTTP scripting
  - https://www.tutorialspoint.com/unix_commands/curl.htm
- Or the Unix / Linux **wget** command
  - http://www.tutorialspoint.com/unix_commands/wget.htm
- Or, the PostMan utility:
  - https://www.getpostman.com/postman

Questions ?

Exercise 1 (Write your name and x.500 id on paper, add your own answers and hand in at end of class)

An HTTP 1.1. Compliant Python webserver is running on the host computer:

`csel-kh1262-11.cselabs.umn.edu.`

The server was executed (i.e., run) from the `/webserver` folder (directory) (which has world executable permissions) and is listening on port `9004`. The webserver pecontents of `/webserver` folder (directory) are as follows:

```
-rw------- 1 x500user CSEL-student 3275 Oct 3 07:25 server.py
-rw-r--r-- 1 x500user CSEL-student 1261 Oct 4 17:42 schedule.html
-rw-r--r-- 1 x500user CSEL-student  368 Oct 4 21:40 main.js
-rw-r--r-- 1 x500user CSEL-student 2561 Oct 4 17:42 my schedule.html
```

What HTTP 1.1 message response code will be sent to the client/browser after the following requests are sent by the client/browser:

```
GET /webserver/schedule.html HTTP/1.1
Host: csel-kh1262-11.cselabs.umn.edu
Accept: */*

DELETE /webserver/main.js HTTP/1.1
Host: csel-Kh1262-11.cselabs.umn.edu
Accept: */*

HEAD /webserver/my schedule.html HTTP/1.1
Host: csel-kh1262-11.cselabs.umn.edu
Accept: */*
```

JavaScript Object Notation

- JavaScript Object Notation (JSON)
- References – Chapter 10.3.3 Sebesta
  - https://www.w3schools.com/js/js_json_intro.asp
  - https://www.w3schools.com/js/js_json.asp
  - www.json.org

JSON

- Lightweight data interchange format
- Self-describing – human readable and writeable
- It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999.
- JSON is a text format that is completely language independent **BUT**
- It uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others.

- JSON is built on two structures:
  - A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
  - An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence

JSON Values Can Be:

- A number (integer or floating point)
- A string (in double quotes)
- A Boolean (true or false)
- An array (in square brackets)
- An object (in curly braces)
- null

JSON Objects / JSON Arrays

- JSON objects are written inside curly braces.
- Just like in JavaScript, objects can contain multiple name/values pairs:
  - e.g., {"firstName":"John", "lastName":"Doe"}
- JSON arrays are written inside square brackets.

- As in JavaScript, an array can contain multiple objects:
- {"employees":[
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter", "lastName":"Jones"}
  ]}
- The object "employees" is an array containing three objects. Each object is a record of a person (with a first name and a last name).

Have you used JSON in this Course Berfore?

Where?

JSON Uses JavaScript Syntax

Example:

```
var employees = [
   {"firstName":"John", "lastName":"Doe"},
   {"firstName":"Anna", "lastName":"Smith"},
   {"firstName":"Peter", "lastName": "Jones"}
];
```

The first entry in the JavaScript object array can be accessed as follows:

```
employees[0].firstName + " " + employees[0].lastName;
```

The content returned will be:
John Doe

Data in the array can be modified as follows:

```
employees[0].firstName = "Gilbert";
```

Creating JSON Objects from a string

The following creates a JavaScript string containing JSON syntax:

```
var text = '{ "employees" : [' +
'{ "firstName":"John" , "lastName":"Doe" },' +
'{ "firstName":"Anna" , "lastName":"Smith" },' +
'{ "firstName":"Peter" , "lastName":"Jones" } ]}';
```

The JavaScript function **JSON.parse(*text*)** can be used to convert text in a JSON format into a JavaScript object:

```
E.g., var obj = JSON.parse(text);
```

Example – JSON to JavaScript Objects

jsonex1.html

An HTML and JAVASCRIPT example that starts with text stored in JSON notation

Converts the text to a JavaScript Object

Displays the contents of the Object

Example – JSON to JavaScript Objects

```
<!DOCTYPE html>
<html>
<body>

<h2>JSON Object Creation in JavaScript</h2>

<p id="demo"></p>

<script>
var text = '{"name":"President Trump","streetaddress":"1600 Pennsylvania Ave.","phone":"202 4561414"}'

var obj = JSON.parse(text);


document.getElementById("demo").innerHTML =
obj.name + "<br>" +
obj.streetaddress + "<br>" +
obj.phone;
</script>

</body>
</html>
```

jsonex1.html

Example 2: JSON to JavaScript Arrays

jsonex2.html

An HTML and JAVASCRIPT example that starts with text stored in JavaScript Array notation

Converts the text to a JavaScript Array

Displays the contents of the Array

Example 2: JSON to JavaScript Arrays

```
<!DOCTYPE html>
<html>
<body>

<h2>JSON Array Creation in JavaScript</h2>
		<p id="result"></p>

		<script>
		var nums = '["200","400","600","800"]';
		var anarray = JSON.parse(nums);

		var sum = 0;
		for (i = 0; i < anarray.length; i++) {
			sum += parseInt(anarray[i]);
		}
		document.getElementById("result").innerHTML = sum;
		</script>
</body>
</html>
```

jsonex2.html

JSON – Evaluates to JavaScript Objects

- The JSON format is syntactically identical to the code for creating JavaScript objects.
- Because of this similarity, instead of using a parser (like XML, which we will see later in the course, does), a JavaScript program can use standard JavaScript functions to convert JSON data into native JavaScript objects.

Example of Reading a JSON file using JavaScript and XMLHttpRequest()

- http://www-users.cs.umn.edu/~chal0006/JSON/JSONHttpRequest.html

Actually, this is an example of AJAJ

- AJAJ is Asynchronous JavaScript and JSON

AJAX

- Not a cleaning product that is stronger than dirt
- AJAX = Asynchronous JavaScript and XML
- Enables the implementation of more efficient web pages

AJAX

- Enables web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
- Web pages that do not use AJAX reload the entire page if any content on the page changes

AJAX – Based on Internet Standards

- Uses a combination of:
  - XMLHttpRequest object (to exchange data asynchronously with a server)
  - JavaScript/DOM (to display/interact with the information)
  - CSS (to style the data)
  - XML (often used as the format for transferring data) – but can be JSON or just plain text

Who uses AJAX?

- Google (Gmail, Maps and Suggest)
- Facebook (tabs)
- Youtube

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

Next Time

- More on AJAX, Node.js

- Readings – at the beginning of this bunch-o-lecture slides
- I'll update schedule and notify you of the update