

Processing JSON with JavaScript



IT 4403 Advanced Web and Mobile
Applications

Jack G. Zheng
Fall 2019



KENNESAW STATE
UNIVERSITY



Overview

- JSON basics
- Reading JSON file/content using JavaScript/jQuery

Data Transfer/Exchange



- Data can be represented in memory in various models or structures (relation, object, etc.), but when it is transferred between distributed systems and computers, it needs to be represented in plain text files
 - The process is usually called serialization
<https://en.wikipedia.org/wiki/Serialization>
- Popular formats
 - Comma Separated Values (CSV)
https://en.wikipedia.org/wiki/Comma-separated_values
 - eXtensible Markup Language (XML)
 - JavaScript Object Notation (JSON)

JSON



- JSON (JavaScript Object Notation) is a lightweight data format.
 - It is easy for humans to read and write.
 - It is easy for machines to parse and generate.
 - It is based on a subset of the JavaScript Programming Language.

JSON Syntax



- JSON is built on two structures:
 - A collection of name/value pairs.
 - An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.
- Find detailed definition at <http://www.json.org>
 - An object is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).
 - An array is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).
 - A value can be a string in double quotes, or a number, or true or false or null, or an object or an array. These structures can be nested.

Example: A Sample JSON Document



Notice how data are organized in hierarchical { } for an object.

{

"Instructor":

{

"@HireDate": "2011-08-01",

"FirstName": "Jack",

"LastName": "Zheng",

"Title": "Assistant Professor"

}

}

There is no attribute concept in JSON. "@CRN" uses @ to denote as an attribute converted from XML. This is not standard.

All items are in name:value pairs. We use { } for a complex value type, such as the "Instructor" value

Items are separated by comma

Names are always enclosed by " "

JSON Format Examples



- Examples are provided in D2L as well as online <http://it4203.jackzheng.net/demo/json/>

Files (Examples)	Explanation
instructor1.json	A simple example with one data item and one level of data. Note there is no attribute node in JSON, so using @ to indicate an attribute node from its corresponding XML format – this is not a standard.
instructor2a.json instructor2b.json	The document consists of multiple data items (records) with the same structure (three instructors). In instructor2b.json, a root element of Instructors is not required actually. JSON can start with an array directly []
instructor3.json	A mixture of simple elements (Instructor data) and more complex elements (Course). In JSON, number type values do not have quotes. This example will be the main one to demonstrate how JSON content is read by JavaScript/jQuery later.
instructor4.json	Some variation of the last example, with additional level of elements (Teaching). This additional level is not required and is more verbose. But you do get some documents like this from time to time.

Free JSON Tools for Developers



- Recommended viewer, converter, validator
 - <http://jsonformatter.org>
 - <http://codebeautify.org/jsonviewer>
- JSON Viewer browser plugin
 - JSONView
<https://chrome.google.com/webstore/detail/jsonview/chklaanhfefbnpoihckbnefhakgolnmc> (good to provide element path)
 - JSON Formatter <http://goo.gl/ZDLWY0>
- Other JSON viewer, validator, formatter
 - <http://jsonlint.com/> (also try the pro version)
 - <http://jsonprettyprint.net/>
- Converter
 - <http://www.utilities-online.info/xmltojson>
 - <http://www.freeformatter.com>

XML vs. JSON



	JSON	XML
Extended From	JSON is extended from JavaScript.	XML is extended from SGML : “Standard Generalized Markup Language“.
Purpose	JSON is one type of text-based format or standard for interchanging data i.e. human readable. JSON is developed by “Douglas Crockford”.	XML is a Markup Language having format that contains set of rules for the encoding the documents which is readable for both human & machine. XML is developed by W3C:“World Wide Web Consortium “.
Syntax	JSON syntax is lighter than XML as JSON has serialized format of data having less redundancy. JSON does not contain start and end tags.	XML is not so lighter as JSON as having start and end tags and it takes more character than JSON to represent same data.
Speed	JSON is light – weighted in compare to XML as it has serialized format and so faster also.	XML is not so light weighted as JSON.
Support of Data Type & Array	JSON supports datatype including integer and strings, JSON also supports array.	XML does not provide any data type so needs to be parsed into particular datatype. No direct support for array also.
Object Support	JSON has support of native objects.	XML can get support of objects through mixed use of attributes & elements.
Comments	JSON does not support Comments	XML supports comments.
Namespace	JSON does not have support for Namespaces.	XML supports Namespaces.
Mapping	JSON is data oriented and can be mapped more easily.	XML is document oriented and needs more effort for mapping.

Some Syntactical Differences



	JSON	XML
Tags	JSON does not contain start and end tags.	XML is not so lighter as JSON as having start and end tags and it takes more character than JSON to represent same data.
Attributes	No attributes	Has attribute node
Root element	JSON may not start with a root element. It can start with an array.	There has to be a root element
Support of Data Type & Array	JSON supports datatype including integer and strings, JSON also supports array.	XML does not provide any data type so needs to be parsed into particular datatype. No direct support for array also.
Namespace	JSON does not have support for Namespaces.	XML supports Namespaces.

For more comparison, see:

- <http://www.json.org/xml.html>
- <https://www.udemy.com/blog/json-vs-xml/>
- <http://www.infoq.com/news/2006/12/json-vs-xml-debate>
- https://blogs.oracle.com/xmlorb/entry/analysis_of_json_use_cases



Reading JSON Content

- JSON can be processed (read) by both server side languages (PHP, C#) and client side languages (JavaScript, jQuery)
- We will focus on the client side since this course is more about frontend development.
- Reading JSON
 - This is very simple in JavaScript as JSON is part of JavaScript.

General Explanation of Examples



- Some examples need the support of PHP. So please place these examples in a web server (like Azure) and request them from a web server.
- Make sure your web server serve the file types of JSON. XML is usually supported by default. If not supported, you will get a file not found error. Configure the server MIME setting if necessary.
- I tried to make the examples self-explanatory. Ask questions in class or in discussion boards.



Basics to Read JSON

- Use object notation to retrieve values
- Two ways to retrieve a value (assuming “json” is the JS variable to represent a JSON object)
 1. `json.Instructor`
 2. `json[“Instructor”]` ←
- Read from an array
 - `json.Instructor[0]`

Use this method to deal with special characters in names like space, : @, #, -, _, and dynamic field name
<http://stackoverflow.com/questions/4925760/selecting-a-json-object-with-a-colon-in-the-key>

Reading JSON Examples



Files (Examples)	Explanation
read-json-1.html	JSON content is stored as a native JS object. Then use jQuery to read the data and display it on the page.
read-json-2.html	This example is similar to the first one except that JSON content is stored as a string in a JS variable.
instructor3.json	Sample JSON file used with the examples below
read-json-3.html	This time we use an AJAJ (AJAX) way to load the file without the help of PHP. Your server needs to be configured to be able to serve JSON type files. This is just for your reference if you have more time. We will cover more details on this type of techniques in the future module.
read-json-4.php	For this example, we try to load the file at the server side and dump all its content to a JavaScript variable; we do this using PHP. We then use traditional JS to read JSON content.
read-json-5.php	This is very similar to the last example, except we use jQuery. The way how it refers to different parts of the JSON content is the same.

Examples are provided in D2L as well as online
<http://it4203.azurewebsites.net/demo/json/>



JSON string vs. JSON object

- JSON string

```
var person='{ "name":"John", "age":30, "city":"New York"}';
```

- JSON object

The difference is the quotation mark.

```
var person={ "name":"John", "age":30, "city":"New York"};
```

- JavaScript cannot directly process a JSON string, often the case when JSON content is passed from another application or web.
- Use the following two methods to convert between string and object
 - JSON.parse https://www.w3schools.com/js/js_json_parse.asp
 - JSON.stringify https://www.w3schools.com/js/js_json_stringify.asp

A Note on Azure



- Create a “web.config” file with the following lines to enable JSON serving.
- <https://blogs.msdn.microsoft.com/africaapps/2013/06/07/how-to-serve-static-json-files-from-a-windows-azure-website/>

```
<?xml version="1.0"?>
<configuration>
  <system.webServer>
    <staticContent>
      <mimeMap fileExtension=".json"
        mimeType="application/json" />
    </staticContent>
  </system.webServer>
</configuration>
```

Error Handling and Debugging



- Common errors
 - Spelling mistakes
 - Wrong hierarchy
 - Missing element
 - Null value
 - Special character
- Demo: handling common errors

Learning Materials and Resources



- Core learning resources
 - A brief intro: <https://beginnersbook.com/2015/04/json-tutorial/>
 - A really good video tutorial: <https://www.youtube.com/watch?v=iiADhChRriM>
 - Try some live practice: http://www.w3schools.com/js/js_json.asp
 - https://www.w3schools.com/js/js_json_intro.asp: from “Intro” to “Arrays” section on the left menu (you may read the rest but we are covering them in other modules).
- Additional resources and readings
 - Complete reference: <http://www.json.org>
 - JSON vs. XML: <https://www.udemy.com/blog/json-vs-xml/>
 - Some more complete tutorials:
 - <https://www.w3resource.com/JSON/introduction.php>
 - <https://restfulapi.net/introduction-to-json/>
 - <https://www.tutorialspoint.com/json/>