COMP30680
Web Application Development

JSON

David Coyle
d.coyle@ucd.ie

What is JSON?

JSON stands for **J**ava**S**cript **O**bject **N**otation.

It is a lightweight data-interchange format that defines a syntax for storing and exchanging data.

JSON is language independent. It uses JavaScript syntax, but the JSON format is text only.
It can be read and used as a data format by any programming language.

A common use of JSON is to read data from a web server, and display the data in a web page.

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

JSON & XML

JSON has many similarities to XML but is designed to be easier to use.

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

```
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName> <lastName>Jones</lastName>
    </employee>
</employees>
```

JSON vs XML

Similarities:

- Both JSON and XML are "self describing" (human readable)
- Both JSON and XML are hierarchical (values within values)
- Both JSON and XML can be parsed and used by lots of programming languages
- Both JSON and XML can be fetched with an XMLHttpRequest

Differences:

- JSON doesn't use end tags
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

Using XML

- Fetch an XML document
- Use the XML DOM to loop through the document
- Extract values and store in variables

Using JSON

- Fetch a JSON string
- JSON.Parse the JSON string

But the most important difference for us:

XML has to be parsed with an XML parser, JSON can be parsed by a standard JavaScript function.

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 does), a JavaScript program can use standard JavaScript functions to convert JSON data into native JavaScript objects.

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

<script>
var text = '{"name":"John Johnson","street":"Oslo West 16","phone":"555 1234567"}';

var obj = JSON.parse(text);

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

Parse →

Use the JavaScript object →

JSON Syntax

JSON syntax is derived from JavaScript object notation syntax.

- Data is in name/value pairs
  A name/value pair consists of a name (in double quotes), followed by a colon, followed by a value:

```
"firstName":"John"
```

NOTE: JSON names require double quotes. JavaScript names don't.

- Data is separated by commas

```
{"firstName":"John", "lastName":"Doe"}
```

- Curly braces hold objects
- Square brackets hold arrays

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

Just like JavaScript, JSON objects can contain multiple name/values pairs.
Similarly a JSON array can contain multiple objects.

JSON values

JSON format supports the following data types:

Type	Description
Number	double- precision floating-point format in JavaScript
String	double-quoted Unicode with backslash escaping
Boolean	true or false
Array	an ordered sequence of values
Value	it can be a string, a number, true or false, null etc
Object	an unordered collection of key:value pairs
Whitespace	can be used between any pair of tokens
null	empty

JSON example – reading a txt file.

The code for this example is in the zip file for this lecture.

See:
readtxt_v1.html and **readtxt_v2.html**.

In the v2 file the array declaration has been moved to a separate txt file and is read in using an **XMLHttpRequest**. This is a common method to read files that contain xml or JSON.

The example is from w3schools:
http://www.w3schools.com/json/json_http.asp

```
<div id="id01"></div>

<script>

var myArray = [
{
"display": "JavaScript Tutorial",
"url": "http://www.w3schools.com/js/default.asp"
},
{
"display": "HTML Tutorial",
"url": "http://www.w3schools.com/html/default.asp"
},
{
"display": "CSS Tutorial",
"url": "http://www.w3schools.com/css/default.asp"
}
];

myFunction(myArray);

function myFunction(arr) {
    var out = "";
    var i;
    for(i = 0; i < arr.length; i++) {
        out += '<a href="' + arr[i].url + '">' +
        arr[i].display + '</a><br>';
    }
    document.getElementById("id01").innerHTML = out;
}

</script>
```

readtxt_v1.html

myArray is defined within the script tag in the JS file.

A function loops through the array and displays the contents.

```
<script>

var xmlhttp = new XMLHttpRequest();
var url = "myTutorials.json";

xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
        var myArr = JSON.parse(xmlhttp.responseText);
        myFunction(myArr);
    }
};

xmlhttp.open("GET", url, true);
xmlhttp.send();


function myFunction(arr) {
    var out = "";
    var i;
    for(i = 0; i < arr.length; i++) {
        out += '<a href="' + arr[i].url + '">' +
        arr[i].display + '</a><br>';
    }
    document.getElementById("id01").innerHTML = out;
}
</script>
```

readtxt_v2.html

An array is now defined in an external JSON file and read in using an XMLHttpRequest. MyArray is created by parsing the result.

The function remains the same.

JSON example – getting data from a web server

The code for this example is also in the zip file for this lecture.

See:
readFromServer.html

In this case the JSON data is ‘read’ from the following url:
http://www.w3schools.com/js/customers_mysql.php

By opening the url in a browser you can see the JSON text.

Again the example is from w3schools:
http://www.w3schools.com/json/json_example.asp

JSON resources

Online viewer and editor:
http://www.jsoneditoronline.org/

Sample files:
Color data: **colorData.json**
Customer data example: **customerData.json**
YouTube response: **youtube.json**
OpenWeatherMap: openWeatherMap.json - See also http://openweathermap.org/forecast5

A quick guide:
http://www.tutorialspoint.com/json/json_quick_guide.htm

Questions, Suggestions?

Next:

Restful APIs