

Copyright © 2015 The Learning House.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of The Learning House. For permission requests, write to The Learning House, addressed “Attention: Permissions Coordinator,” at the address below.

The Learning House
427 S 4th Street #300
Louisville KY 40202

JavaScript Object Notation (JSON)

.NET Cohort

Coding Bootcamp

What is JSON?

When we transmit data over the internet, we have a few choices: XML and JSON

XML

```
<student>
  <name>Eric</name>
  <age>19</age>
  <school>
    <name>University of Toledo</name>
    <location>Toledo, OH</location>
  </school>
</student>
```

JSON

```
{
  "name": "Eric",
  "age": 19,
  "school": {
    "name": "University of Toledo",
    "location": "Toledo, OH"
  }
}
```

Why JSON?

JSON is more terse than XML, and JavaScript provides an `eval()` method that creates a JavaScript object directly from JSON data, which makes it much easier to work with than converting XML to a JavaScript object.

Data Structures

JSON supports simple values: 5

JSON supports objects: { “property” : value }

JSON supports arrays: { “colors” : [“red”, “blue”, “green”] }

An array of book objects with arrays of authors?

Sure, why not?

```
[{
  "title": "Professional JavaScript",
  "authors": [
    "Nicholas C. Zakas"
  ],
  edition: 3,
  year: 2011
},
{
  "title": "Professional Ajax",
  "authors": [
    "Nicholas C. Zakas",
    "Jeremy McPeak",
    "Joe Fawcett"
  ],
  edition: 2,
  year: 2008
}];
```

The JSON object

ECMAScript 5 created a native global object called JSON. The object is supported in IE 8+, Firefox 3.5+, Safari 4+, Chrome, and Opera 10.5+.

If you need to support older browsers, you can download a JavaScript shim file or use jQuery instead.

The JSON object has two methods, `stringify()` and `parse()`. `stringify()` turns a JavaScript object into JSON, and `parse` turns JSON data into a JavaScript value.

Example of stringify() and parse()

```
var movie = {  
  title: "Ghostbusters",  
  actors: [  
    "Bill Murray",  
    "Dan Aykroyd"  
  ],  
  year: 1985  
};  
  
var jsonText = JSON.stringify(movie);  
// value of jsonText is:  
// {"title":"Ghostbusters","authors":["Bill Murray","Dan Aykroyd"],"year":1985}  
  
// convert into another movie object  
var movie2 = JSON.parse(jsonText);
```


Filtering stringify()

If you only want some properties to be turned into JSON, you can list them in an array as a parameter like so:

```
var movie = {  
  title: "Ghostbusters",  
  actors: [  
    "Bill Murray",  
    "Dan Aykroyd"  
  ],  
  year: 1985  
};  
  
var jsonText = JSON.stringify(movie, ["title", "year"]);
```

toJSON()

If you want to do custom serialization, just define a toJSON function on your object like so:

```
var movie = {
  title: "Ghostbusters",
  actors: [
    "Bill Murray",
    "Dan Aykroyd"
  ],
  year: 1985,
  toJSON: function () {
    return this.title;
  }
};

// jsonText will only contain "Ghostbusters"
var jsonText = JSON.stringify(movie);
```

Customized Parsing

We can also provide a function as a second parameter to parse.

The function will be called on each key value pair in the JSON data.

In this example we convert the `releaseDate` to a JavaScript date time (it would default to a string otherwise)

```
var movie = {
  title: "Ghostbusters",
  actors: [
    "Bill Murray",
    "Dan Aykroyd"
  ],
  year: 1984,
  releaseDate: new Date(1984, 6, 7)
};

var jsonText = JSON.stringify(movie);

// convert into another movie object, but instead of a
// string date, create a JavaScript date
var movie2 = JSON.parse(jsonText, function(key, value){
  if (key == "releaseDate"){
    return new Date(value);
  } else {
    return value;
  }
});
```

Summary

JSON is quickly replacing XML as the de-facto data sharing format. You really need to be able to read and manipulate it to be successful as a web developer.