

# Json Reference

By John Bentley

## Introduction

### Key

This is text is for general narration.

The next paragraph.

This is a rule to follow.

```
This contains any code or other examples.
```

This justifies the rule.

| This is a quote that also justifies the rule.

*This is the source for the rule (if there are not multiple source that refer to different parts of the example, justification, or justification quote).*

Another rule to follow.

## what is it?

JavaScript Object Notation (JSON) is a data interchange format. It is stored in a text file. It is human and machine readable. Therefore it can be sent over the internet and used as an alternative to XML.

JSON is a subset of JavaScript.

## why would you use it?

It's reason for being is to enable simple and obvious data referencing in AJAX applications.

During usual AJAX applications JavaScript is used to traverse the DOM of the XML returned from the server. This can lead to cumbersome referencing.

If JSON is returned, instead of XML, then the data can be referenced using in a more simple and obvious syntax.

You don't have to use JavaScript to process a JSON text file.



```

    "ceiling": 37000
  },
  {
    "code": "BE58",
    "manufacturer": "Beechcraft",
    "model": "Baron 58",
    "seats": 6,
    "engineType": "propeller",
    "ceiling": 20688
  }
]
}}

```

### No Root Format.

```

{
  "type": [
    {
      "code": "B734",
      "manufacturer": "Boeing",
      "model": "737-400",
      "seats": 168,
      "engineType": "jet",
      "ceiling": 37000
    },
    {
      "code": "BE58",
      "manufacturer": "Beechcraft",
      "model": "Baron 58",
      "seats": 6,
      "engineType": "propeller",
      "ceiling": 20688
    },
    ...
  ]
}

```

### XML Equivalent (to either JSON format).

```

<?xml version="1.0"?>
<aircraft>
  <type code="B734">
    <manufacturer>Boeing</manufacturer>
    <model>737-400</model>
    <seats>168</seats>
    <engineType>jet</engineType>
    <ceiling>37000</ceiling>
  </type>
  <type code="BE58">
    <manufacturer>Beechcraft</manufacturer>
    <model>Baron 58</model>
    <seats>6</seats>
    <engineType>propeller</engineType>
    <ceiling>20688</ceiling>
  </type>
</aircraft>

```

## Coding

### JavaScript

Same domain, explicit root format.

```

function ajaxOnSuccess() {
  //Normal Ajax handling:

```

```
//var xmlDocDocument = Xhr.responseXML;

var jsonObject = eval('(' + Xhr.responseText + ')');

// The JSON file is in
// explicit root object format.
var aircraft = jsonObject.aircraft;

// Application Specific Code Bellow
var xhtml = "";
xhtml += "<table>";
xhtml += "<thead><tr>";
xhtml += "<td>Manufacturer</td>";
xhtml += "<td>Model</td>";
xhtml += "</tr></thead>";
xhtml += "<tbody>";

for (var i = 0; i < aircraft.type.length; i++) {
    xhtml += "<tr>";
    xhtml += "<td>" + aircraft.type[i].manufacturer + "</td>";
    xhtml += "<td>" + aircraft.type[i].model + "</td>";
    xhtml += "</tr>";
}
xhtml += "</tbody>";
xhtml += "</table>";

// Insert into the web page.
document.getElementById("output").innerHTML = xhtml;
}
```

### Same domain, no root format

```
function ajaxOnSuccess() {
    var aircraft = eval('(' + Xhr.responseText + ')');
    ...
}
```

### Cross domain using JSON parser. Use the parser cross domain for security reasons.

```
function ajaxOnSuccess() {
    var aircraft = JSON.parse(Xhr.responseText);
    ...
}

// Using the parser in the other
// direction.
var myJSONText = JSON.stringify(aircraft);
```

(Crockford [2010] 2019, \*JSON in JavaScript\*, <https://github.com/douglascrockford/JSON-js>)  
<https://github.com/douglascrockford/JSON-js>

## Php

### Convert JSON String to Array or Object.

```
<?php
// JSON string
$someJSON = '{"name":"Jonathan Suh","gender":"male"}, {"name":"William Philbin","gender":"male"}, {"name":"Allison McKinnery","gender":"female"}';

// Convert JSON string to Array
$someArray = json_decode($someJSON, true);
print_r($someArray); // Dump all data of the Array
echo $someArray[0]["name"]; // Access Array data

// Convert JSON string to Object
$someObject = json_decode($someJSON);
```

```
print_r($someObject); // Dump all data of the Object
echo $someObject[0]->name; // Access Object data
?>
```

(Suh n.d., "Convert and Loop through JSON with PHP and JavaScript Arrays/Objects", Accessed 2019-08-16.  
<https://jonsuh.com/blog/convert-loop-through-json-php-javascript-arrays-objects/>) <https://jonsuh.com/blog/convert-loop-through-json-php-javascript-arrays-objects/>

### Convert JSON String to Object from File.

```
// Using No Root Format, as above.

<pre><?php
$jsonString = file_get_contents('jsonAircraftNoRoot.json');

// Converts JSON string to Object
$myObject = json_decode($jsonString);
echo htmlspecialchars(print_r($myObject, true), ENT_XML1);
echo $myObject->type[0]->model; // Outputs "737-400"
?></pre>
```

<\\Atlas\Users\John\Documents\Sda\Code\Php\Examples\json\json.php>

Ensure your source JSON file is encoded as UTF8, not UTF with BOM.

Visual Studio Code > Open json file > Status Bar: "UTF-8", not "UTF-8 with BOM"

(Bentley, n.d., "Experimentally Verified") 2019-08-16  
<\\Atlas\Users\John\Documents\Sda\Code\Php\Examples\json\json.php>

## References, Zotero

Crockford, Douglas. [2010] 2019. *\*JSON in JavaScript\**. JavaScript. <https://github.com/douglascrockford/JSON-js>. (November 18, 2010) 2019-08-15.

Suh, Jonathan. n.d. "Convert and Loop through JSON with PHP and JavaScript Arrays/Objects". Jonathan Suh. Accessed 2019-08-16. <https://jonsuh.com/blog/convert-loop-through-json-php-javascript-arrays-objects/>. n.d.

## Document Licence

[Json Reference](#) © 2021 by [John Bentley](#) is licensed under [Attribution-NonCommercial-ShareAlike 4.0 International](#)

