JSON

+4

Published Jun 2, 2022•**Updated Apr 22, 2023**

Contribute to Docs

JavaScript Object Notation (JSON) is a language-independent data format that is readable, writable, and parsable for both humans and machines. JSON is based on the syntax of the third edition of a JavaScript standard known as (ECMAScript). Many programming languages, such as Python, have implemented libraries to parse and generate JSON-formatted data. JavaScript can parse JSON directly with the JSON object.

In addition to JavaScript, JSON uses some conventions from the C-descendent languages, including C , C++, and Java.

JSON is commonly used for transferring data between clients and servers for tasks such as web browsing or form submission. Some companies also use JSON to allow their data to be accessed in other applications via API. Some examples include:

- Google Maps
- Google Auth 2.0 Authentication
- Meta (formerly Facebook) Send API
- Spotify Music Web API
- LinkedIn Profile API

More can be learned about JSON by visiting json.org.

Syntax

```
{
 "propertyOne": "valueOne",
 "propertyTwo": "valueTwo",
 "propertyThree": "valueThree",
}
```

Data is stored in an object, depicted by a pair of curly braces {}, and name-value pairs are separated by a colon :. The pairs themselves are separated by a comma ,. The following are data types that can be used:

- Array: An ordered, comma delimited, list of zero or more elements of one of JSON's data types, enclosed in square brackets.
- Boolean: Either true or false.
- Number: A signed decimal number. JSON makes no distinction between integer and floating point.
- Object: A collection of name-value pairs inside curly brackets.
- String: A sequence of zero or more Unicode characters enclosed in double quotes.
- null: An empty value represented by the word null.

Whitespace (e.g., spaces, tabs, line feeds, and carriage returns) is ignored between names, values, and punctuation. The following four characters are all considered whitespace: space, tab, line feed, and carriage return.

JSON objects cannot contain comments.

Multiple JSON objects can be collected in an array-like sequence:

```
[
 {
  "propertyOne": "valueOne",
  "propertyTwo": "valueTwo",
  "propertyThree": "valueThree",
 },
 {
  "propertyA": "valueA",
  "propertyB": "valueB",
  "propertyC": "valueC",
 }
]
```

A few important points to note about JSON syntax:

- While it is derived from from the JavaScript language, JSON itself is not JavaScript.

- Trailing commas are forbidden.
- Although JavaScript names are not this strict, JSON property names must be in double quotes.

Bad Practices

Below are two points regarding what should be avoided when using JSON format:

- It is not useful to abide by an XML-like, “attribute vs. element” framework. JSON is only about name-value pairs.
- While the nesting of inner-JSON objects is possible, going more than two levels deep could get complicated.

Example

The following JSON example uses one level of nested objects, an array, and each valid data type:

```
{
 "make" : "Chevy",
 "model" : "Silverado",
 "miles" : 27500.5,
 "year" : 2020,
 "owner" : {
  "firstName" : "John",
  "lastName" : "Doe"
  },
 "features" : ["4WD", "Towing Package", "Lift Kit"],
 "lease" : false,
 "customizations": null
}
```