**JSON**

1. **What is JSON?**

Many applications communicate by exchanging JSON instead of XML documents. JSON (JavaScript Object Notation) is a language-independent data format that expresses JSON objects as human-readable lists of properties (name-value pairs).

JSON is commonly used in asynchronous browser/server communication via AJAJ. JSON is also used with NoSQL database management systems such as MongoDB and CouchDB. There are multiple apps from social media web sites such as Twitter, Facebook, LinkedIn, and Flickr use JSON as data-interchange format.

Many developers prefer JSON to XML because they see JSON as being less verbose and easier to read.

1. **JSON Syntax**

The JSON data format presents a JSON object as a brace-delimited and comma-separated list of properties:

{

property1,

property2,

…

propertyN

}

A comma is not placed after the final property. For each property, the name is expressed as a string that’s typically quoted (and by a pair of double quotes). The name string is followed by a colon character, which is followed by a value of a specific type ("name": "JSON", for example)

JSON supports the following six types:

* + *Number*: A signed decimal number that may contain a fractional part and may use exponential (E) notation. JSON doesn’t permit nonnumbers (such as NaN),nor does it make any distinction between integer and floating-point. Furthermore, JSON doesn’t recognize the octal and hexadecimal formats. (Although JavaScript uses a double precision floating-point format for all numeric values, other languages implementing JSON may encode numbers differently.
  + *String:* A sequence of zero or more Unicode characters. Strings are delimited with double quotes and support a back slash escaping syntax
  + *Boolean:* Either of the values true or false
  + *Array*: An ordered list of zero or more values, each of which may be of any type. Arrays use square bracket notation with elements being comma-separated.
  + *Object*: An unordered collection of properties where the names (also called *keys*) are strings. Because objects are intended to represent associative arrays, it’s recommended, although not required, that each key be unique within an object. Objects are delimited with braces and use commas to separate each property. Within each property the colon character separates the key from its value
  + *Null:* an empty value, using the keyword null

Whitespace is allowed and is ignored around or between syntactic elements (values and punctuation). Four specific characters are considered whitespace for this purpose: space, horizontal tab, line feed, and carriage return. Also, JSON doesn’t support comments

Example of a JSON Object that describes a person in terms of a name and an age.



## **Validating JSON Objects**

JSON Schema expresses a schema as a JSON object.

Advantages of JSON Schema:

* + It describes your existing data format.
  + If offers clear, human-readable, and machine-readable documentation.
  + It provides complete structural validation, which is useful for automated testing and validating client-submitted data

The JSON Schema web site focuses on draft version 4 of the JSON Schema specification. This specification is divided into three parts: JSON Schema Core, JSON Schema Validation, and JSON Hyper-Schema.