



What Is JSON?

Jeffrey Erickson | Content Strategist | April 4, 2024



In This Article

[What Is JSON \(JavaScript Object Notation\)?](#)

[JSON Explained](#)

[Why Is JSON Used?](#)

[JSON vs. HTML vs. XML](#)

[JSON Data Types](#)

[Top 5 Use Cases for JSON](#)

[What Is a JSON Document Database?](#)

[Get started with Oracle Autonomous JSON Database for Free](#)

[JSON FAQs](#)

In web and mobile application development, one versatile data format reigns supreme: JavaScript Object Notation, better known as JSON. JSON is a lightweight data interchange format that provides a standardized and efficient way for different systems to exchange data. Thanks to its simplicity, flexibility, and compatibility with popular programming languages, JSON has become a cornerstone technology for building web applications and enjoys avid developer community support.

Here's what you need to know about JSON.

What Is JSON (JavaScript Object Notation)?

JSON (JavaScript Object Notation) is a text-based format for storing and exchanging data that is both human-readable and machine-parsable. As a result, JSON is relatively easy to learn and use.



Talk to sales

troubleshoot. Although JSON has its roots in JavaScript, it has grown into a very capable data format that simplifies data interchange across diverse platforms and programming languages. If you're involved in web development, data analysis, or software engineering, JSON is an important data format to understand.

Key Takeaways

JSON is a popular data format often used by web developers for transferring data between a server and a web application.

Because JSON is text-based, it's easily read by humans and understood by computers.

JSON's language-independent nature makes it an ideal format for exchanging data across different programming languages and platforms.

Many databases have emerged to store and exchange data in JSON.

JSON Explained

JSON is a data format that's commonly used by web developers for transferring data between a server and a web application. Developers often prefer JSON because it simplifies the exchange of data between different technologies. For example, when a user interacts with a web application to make a purchase, the application sends the user's input to the server in JSON format. The server processes the data and sends back a response, also in JSON format, which is then rendered by the web application. This allows for seamless data exchange between client and the server, facilitating fast, dynamic, and interactive web experiences.

Why Is JSON Used?

JSON's language-independent nature makes it an ideal format for exchanging data across different programming languages and platforms. For instance, an application written in Java can easily send JSON data to a Python application. Or a mobile app written in JavaScript can use JSON to communicate with a back-end server written in PHP. Why? Because both systems can parse and generate JSON.

Beyond web development, JSON is often used within an application or an IT system for storing and managing configuration settings. For example, configuration files written in JSON format can contain essential information, such as database connection details, API keys, or user preferences. By storing configuration data in simple, easy-to-read and easy-to-parse JSON files, developers can modify application settings without requiring code changes.

Why Is JSON Popular with Developers?

JSON is popular with developers because it's a flexible format for data exchange that enjoys widespread support in modern programming languages and software systems. It's text based and lightweight and a simple to parse data format, meaning it requires no additional code to understand and interpret the data.

JSON has gained momentum in API programming and web services because it delivers a consistent and easy-to-understand interchange and web service results. It also helps that developers have ready access to a variety of document databases, such as MongoDB and others, that store data in JSON format and require no

additional processing when they exchange data. Popular [relational databases](#) can now handle JSON as a native format, opening even more applications to the benefits of JSON.

JSON vs. HTML vs. XML

There are a number of formats for storing and transmitting data on the web. Three popular choices are JSON, XML, and HTML. JSON and XML are both formats used to store and transmit data, and each has different strengths. HTML is a language used to create the structure of a web page and is often used alongside these data storage formats.

Key Differences

JSON (JavaScript Object Notation) is commonly used for data storage and transfer. JSON is a popular choice for applications that benefit from a simple and easy-to-use data format.

XML (Extensible Markup Language) is a general-purpose markup language similar to JSON that allows for more complex data structures.

HTML (Hypertext Markup Language) is used to create the structure and content of web pages. You will often see it used with other languages, such as CSS (Cascading Style Sheets) and JavaScript, to unify a website's style and add interactivity to its pages.

JSON Data Types

In the context of development, data types are the different types of values that can be stored and manipulated in a programming language. Each data type has its own set of attributes and behaviors. JSON supports several data types, including the following:

1. **Objects.** A JSON object data type is a set of name or value pairs inserted between {} (curly braces). The keys must be strings and separated by a comma and should be unique.
2. **Arrays.** An array data type is an ordered collection of values. In JSON, array values must be type string, number, object, array, Boolean, or null.
3. **Strings.** In JSON, strings are enclosed in double quotation marks, can contain any Unicode character, and are commonly used to store and transmit text-based data, such as names, addresses, or descriptions.
4. **Boolean.** Boolean values are designated as either true or false. Boolean values aren't surrounded by quotes and are treated as string values.
5. **Null.** Null represents a value that is intentionally left empty. When no value is assigned to a key, it can be treated as null.
6. **Number.** Numbers are used to store numerical values for various purposes, such as calculations, comparisons, or data analysis. JSON supports both positive and negative numbers as well as decimal points. A JSON number follows JavaScript's double-precision floating-point format.

JSON Example

JSON works by representing data in a hierarchical fashion, using key-value pairs to store information. JSON data is enclosed in curly braces ({}), with each key-value pair separated by a comma (.). For example, the following JSON represents a person's contact information:

```
{ "name": "Jane Smith", "age": 35, "city": "San Francisco", "phone": "014158889275", "email": "janesmith@sample.com" }
```

In this example, "name," "age," "city," "phone," and "email" are the keys, and "Jane Smith," "35," "San Francisco," "014158889275," and "janesmith@sample.com" are the corresponding values.

Top 5 Use Cases for JSON

JSON is popular and widely used by developers, including those who work with stacks such as MERN, which comprises MongoDB, Express, React, and Node.js, and MEAN, which subs in Angular for React.

- 1. Transferring data between systems.** JSON is ideal for transferring data between different systems and programming languages. For example, say a website database has a customer's mailing address, but it needs to be verified via an API to ensure it's valid. A company can send the address data in the JSON format it's already stored in directly to the address validation service API.
- 2. Generating a JSON object from user-generated data.** JSON is ideal for storing temporary data. For example, temporary data can be user generated, such as a submitted form on a website. JSON can also be used as a serialization data.
- 3. Configuring data for applications.** When developing applications, each needs the credentials to connect to a database as well as a log file path. The credentials and the log file path can be specified in a JSON file to be easily read and used by all the systems involved.
- 4. Simplifying complex data models.** JSON simplifies complex documents down to the components identified as being meaningful and converts the data extraction process to a predictable and human-readable JSON file.
- 5. Configuration files and data storage.** JSON allows for easy manipulation and retrieval of data. Specifically, it supports nested structures, which eases storage of complex and hierarchical data. JSON also supports arrays, making it suitable for storing multiple instances of similar data.

What Is a JSON Document Database?

JSON's popularity among developers has spawned a number of highly capable databases that are dedicated to the data format, including both [SQL](#) and [NoSQL](#) databases.

NoSQL document databases store data directly in JSON format without requiring additional processing. Popular NoSQL databases, such as MongoDB, Redis, and Couchbase also support nesting, references, and arrays, which makes a JSON database easier to maintain. In recent years, document databases have evolved to offer advantages such as flexible schemas and improved scalability and performance. With their support for flexible and dynamic data structures, these databases are well-suited for storing semi-structured data, such as text documents, images, or social media feeds.

Widely used SQL databases, such as [Oracle Database](#), now offer JSON as a data type, allowing developers to work with JSON without having to add a specialized JSON database into their projects. This gives dev teams the well-established benefits of SQL as well as the ability to work with other data types in a [single database](#), including graph, spatial, REST, blockchain, and relational data.

Get started with Oracle Autonomous JSON Database for Free

Those looking to use JSON to manage their data may want to consider [Oracle Autonomous JSON Database](#), a full-featured, cloud-based document database service that makes it simple to develop JSON-centric applications. Oracle Autonomous JSON Database also offers a wide range of sophisticated database features, including NoSQL-style document APIs via Oracle SODA and Oracle Database API for MongoDB, serverless scaling, high performance ACID transactions, and comprehensive security, with low pay-per-use pricing.

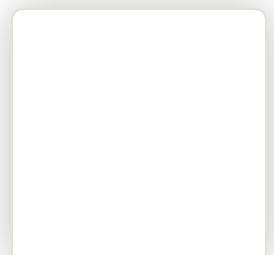
Oracle also offers an easy-to-use database service that automates database management, including capabilities for provisioning, configuring, tuning, scaling, patching, encrypting, and repairing databases. You can get started with Oracle Autonomous JSON Database for free and take advantage of many useful [features that are not found in MongoDB Atlas](#).

JSON has become a simple, versatile, and ubiquitous data format across many domains, including web development, data exchange, configuration management, and data transmission. Its widespread adoption and support among the most popular programming languages have solidified JSON's position as a cornerstone of modern data interchange and communication.

Oracle Autonomous JSON Database is a document database cloud service that makes it simple to develop JSON-centric applications.

[Try Oracle Autonomous Database for free](#)

JSON FAQs



Is JSON a file or code?

JSON is neither a file nor a code. Instead, it’s a simple format used to store and transport data. It is a plain-text format, which allows for easy data interchange between different programming languages. JSON is often used to send data between web applications and servers.

Is JSON is a programming language?

JSON is not a programming language. It is, rather, a lightweight data interchange format. While it was derived from JavaScript, JSON itself does not support functions the way a true programming language does. JSON is simply used for storing and transmitting data between a server and a web application or between different systems.

Is JSON better than XML?

While JSON and XML are both used by developers to store and transfer data between systems, they’re generally used in different circumstances. XML (Extensible Markup Language) is a general-purpose markup language that allows for the creation of complex, hierarchical data structures, while JSON's lightweight and compact nature makes it a better choice for transmitting data over networks—especially in applications where bandwidth is limited or data transmission speed is critical.

Resources for	Why Oracle	Learn	What’s new	Contact us
Careers	Analyst Reports	What is AI?	Oracle Supports Ukraine	US Sales: +1.800.633.0738
Developers	Cloud Economics with Microsoft Azure	What is Cloud Computing?	Oracle Cloud Free Tier	How can we help?
Investors	vs. AWS	What is Cloud Storage?	Cloud Architecture Center	Subscribe to emails
Partners	vs. Google Cloud	What is HPC?	Cloud Lift	Events
Startups	vs. MongoDB	What is IaaS?	Oracle Support Rewards	News
Students and Educators		What is PaaS?	Oracle Red Bull Racing	OCI Blog

