

Node.js Write JSON Object to File – Example Script

Node.js Write JSON Object to File – In this article, we shall learn to write a JSON Object to a local file.

Node.js Write JSON Object to File

To write a JSON Object to a local file, following is a step-by-step guide :

1. Stringify JSON Object. Use `JSON.stringify(jsonObject)` to convert JSON Object to JSON String.
2. Write the stringified object to file using `fs.writeFile()` function of **Node FS** module.

Complete Node.js Script File

In the following Nodejs script, we have JSON data stored as string in variable `jsonData`. We then used `JSON.parse()` function to JSONify the string. So now we have a JSON object. Until now we simulated the situation where you have obtained or created a JSON object.

We would like to save this JSON object to a file.

To save the JSON object to a file, we stringify the json object `jsonObj` and write it to a file using Node FS's `writeFile()` function.

nodejs-write-json-object-to-file.js

```
// file system module  
to perform file
```

```

// file system module to perform file operations
const fs = require('fs');

// json data
var jsonData = '{"persons":[{"name":"John","city":"New York"}, {"name":"Phil","city":"Ohio"}]}';

// parse json
var jsonObj = JSON.parse(jsonData);
console.log(jsonObj);

// stringify JSON Object
var jsonContent = JSON.stringify(jsonObj);
console.log(jsonContent);

fs.writeFile("output.json", jsonContent, 'utf8', function (err) {
  if (err) {
    console.log("An error occurred while writing JSON Object to File.");
    return console.log(err);
  }

  console.log("JSON file has been saved.");
});

```

Useful Link – To access elements of JSON Object, refer [Node.js Parse JSON](#).

Run the above program in Terminal with node command

Node.js Script Terminal Output

```

$ node nodejs-write-
icon-object-to-file.js

$ node nodejs-write-json-object-to-file.js
{ persons:
  [ { name: 'John', city: 'New York' },
    { name: 'Phil', city: 'Ohio' } ] }
{"persons":[{"name":"John","city":"New York"}, {"name":"Phil","city":"Ohio"}]}
JSON file has been saved.

```

A Big Note

In the above program, you might have observed that both `jsonData` and `jsonContent` when logged to console result in same output. This is because when the JSON Object is logged to console, `toString` method is called implicitly. But if you attempt to write the JSON object to a file directly without prior `Stringify`, it results in `[Object Object]` written to file.

Concluding this [Node.js Tutorial](#) – Node.js Write JSON Object to File, we have learned to write a JSON Object

to a file using `JSON.stringify()` function and `FS.writeFile()` function.

Node.js

- [Node.js Tutorial](#)

Get Started With Node.js

- [Install Node.js Ubuntu Linux](#)

- [Install Node.js Windows](#)

- [Node.js - Basic Example](#)

- [Node.js - Command Line Arguments](#)

- [Node.js - Modules](#)

- [Node.js - Create a module](#)

- [Node.js - Add new functions to Module](#)

- [Node.js - Override functions of Module](#)

- [Node.js - Callback Function](#)

- [Node.js - forEach](#)

Express.js

- [Express.js Tutorial](#)

- [What is Express.js?](#)

- [Express.js Application Example](#)

- [Install Express.js](#)

- [Express.js Routes](#)

- [Express.js Middleware](#)

- [Express.js Router](#)

Node.js Buffers

- [Node.js Buffer - Create, Write, Read](#)

- [Node.js Buffer - Length](#)

- [Node.js - Convert JSON to Buffer](#)

- [Node.js - Array to Buffer](#)

Node.js HTTP

- [Node.js - Create HTTP Web Server](#)

- [Node.js - Redirect URL](#)

Node.js MySQL

- [Node.js MySQL](#)

‡ Node.js MySQL - Connect to MySQL Database

‡ Node.js MySQL - SELECT FROM

‡ Node.js MySQL - SELECT WHERE

‡ Node.js MySQL - ORDER BY

‡ Node.js MySQL - INSERT INTO

‡ Node.js MySQL - UPDATE

‡ Node.js MySQL - DELETE

‡ Node.js MySQL - Result Object

Node.js MongoDB

‡ Node.js MongoDB

‡ Node.js - Connect to MongoDB

‡ Node.js - Create Database in MongoDB

‡ Node.js - Drop Database in MongoDB

‡ Node.js - Create Collection in MongoDB

‡ Node.js - Delete Collection in MongoDB

‡ Node.js - Insert Documents to MongoDB Collection

‡ MongoError: failed to connect to server

Node.js Mongoose

‡ Node.js Mongoose Tutorial

‡ Node.js Mongoose - Installation

‡ Node.js Mongoose - Connect to MongoDB

‡ Node.js Mongoose - Define a Model

‡ Node.js Mongoose - Insert Single Document to MongoDB

‡ Node.js Mongoose - Insert Multiple Documents to MongoDB

Node.js URL

‡ Node.js - Parse URL parameters

Node.js FS (File System)

‡ Node FS

‡ Node FS - Read a File

‡ Node FS - Create a File

‡ Node FS - Write to a File

‡ Node FS - Append to a File

‡ Node FS - Rename a File

‡ Node FS - Delete a File

‣ [Node FS Extra - Copy a Folder](#)

Node.js JSON

‣ [Node.js Parse JSON](#)

‣ [Node.js Write JSON Object to File](#)

Node.js Error Handling

‣ [Node.js Try Catch](#)

Node.js Examples

‣ [Node.js Examples](#)

‣ [Node.js - Handle Get Requests](#)

‣ [Node.js Example - Upload files to Node.js server](#)

Useful Resources

‣ [Node.js Interview Questions](#)

‣ [How to Learn Programming](#)