import JSON files in ES modules (Node.js)

```
// An import assertion in a static import
import info from `./package.json` assert { type: `json` };

// An import assertion in a dynamic import
const { default: info } = await import("./package.json", {
  assert: {
    type: "json",
  },
});
```

Option 1: Read and parse JSON files yourself

The Node.js documentation advises to use the `fs` module and do the work of reading the files and parsing it yourself.

```
import { readFile } from 'fs/promises';

const json = JSON.parse(

  await readFile(

    new URL('./some-file.json', import.meta.url)

  )

);
```

Option 2: Leverage the CommonJS `require` function to load JSON files

createRequire allows you to construct a CommonJS require function to use typical CommonJS features such as reading JSON in your Node.js EcmaScript modules.

```
import { createRequire } from "module";

const require = createRequire(import.meta.url);

const data = require("./data.json");
```

Example-1

Method 1: Using require method:

A straightforward way to read a JSON file in a Node JS file is by using the `**require()**` method to include it.

Syntax:

```
const data = require('path/to/file/filename');
```

Example: Create a **users.json** file in the same directory where **index.js** file present. Add following data to the **users.json file** and write the index.js file code:

JSON

```
[
{
        "name": "John",
```

```
        "age": 21,
        "language": ["JavaScript", "PHP", "Python"]
},
{
        "name": "Smith",
        "age": 25,
        "language": ["PHP", "Go", "JavaScript"]
}
]
```

Js code:

```
// Requiring users file
const users = require("./users");

console.log(users);
```

To run the file using the command:

```
node index.js
```

Method 2: Using the fs module:

```
const fs = require("fs");

// Read users.json file
fs.readFile("users.json", function(err, data) {

    // Check for errors
    if (err) throw err;

    // Converting to JSON
    const users = JSON.parse(data);
    console.log(users); // Print users
});
```

Writing to a JSON file

We can write data into a JSON file by using the nodejs **fs** module. We can use **writeFile** method to write data into a file.

Syntax:

```
fs.writeFile("filename", data, callback);
```

Example: We will add a new user to the existing JSON file, we have created in the previous example. This task will be completed in three steps:

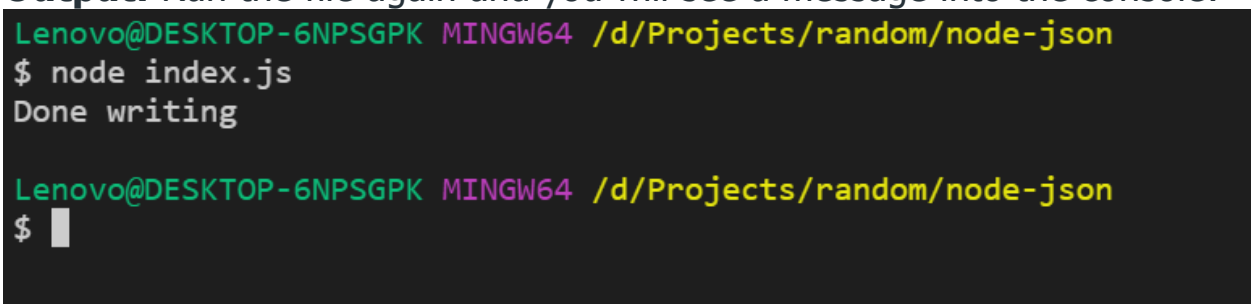
- Read the file using one of the above methods.
- Add the data using `.push()` method.
- Write the new data to the file using `JSON.stringify()` method to convert data into string.

INDEX.js

```
const fs = require("fs");

// STEP 1: Reading JSON file
const users = require("./users");

// Defining new user
let user =
{
    name: "New User",
    age: 30,
    language: ["PHP", "Go", "JavaScript"]
};

// STEP 2: Adding new data to users object
users.push(user);

// STEP 3: Writing to a file
fs.writeFile(
    "users.json",
    JSON.stringify(users),
    err => {
        // Checking for errors
        if (err) throw err;
```

```
    // Success
    console.log("Done writing");
});
```

Output: Run the file again and you will see a message into the console:



Now check your **users.json** file it will looks something like below: