

Node.js & Express.js

NodeJS

- Node.js & Express.js
 - NodeJS
 - Globals
 - Modules
 - Built un Modules
 - Operating System Module (OS)
 - Path Module

NodeJS is an environment to run JavaScript outside of a browser.

Globals

`__dirname` - path to the current directory

`__filename` - current file name

`require` - function to use modules (CommonJS)

`module` - Info about current module (file)

`process` - Info about env where the program is being executed

Modules

Global **module** variable contains information about the corresponding file, it is a JSON object.

```
Module {
  id: '.',
  path:
'C:\\Users\\Jeremy\\Desktop\\code\\FreeCodeCamp\\Backend_And_API\\Nodes_Tutorial_1',
  exports: {},
  filename:
'C:\\Users\\Jeremy\\Desktop\\code\\FreeCodeCamp\\Backend_And_API\\Nodes_Tutorial_1\\app.js',
  loaded: false,
  children: [],
  paths: [
'C:\\Users\\Jeremy\\Desktop\\code\\FreeCodeCamp\\Backend_And_API\\Nodes_Tutorial_1\\node_modules',
'C:\\Users\\Jeremy\\Desktop\\code\\FreeCodeCamp\\node_modules',
'C:\\Users\\Jeremy\\Desktop\\code\\node_modules',
'C:\\Users\\Jeremy\\Desktop\\node_modules',

```

```
'C:\\Users\\Jeremy\\node_modules',  
'C:\\Users\\node_modules',  
'C:\\node_modules'  
]  
}
```

module variable contains a exports object that one can set as needed. Let's say we have a modulus.js file containing a variable we want to share with the app.js file.

In the modulus.js we can set the exports property as follow

```
// modulus.js file  
const toBeShared = "I'm to be shared"  
module.exports = {toBeShared}
```

Then we can access this variable from the app.js with :

```
// app.js file  
const toBeShared = require('./modulus.js')  
console.log(toBeShared)
```

Our module is now accessible as an object

```
{ toBeShared: "I'm to be shared" }
```

Other flavour of exports :

```
// Still in modulus.js file  
  
module.exports.items = ['item1', 'item2']  
const person = {  
  firstname: "John",  
  lastname: "Snow",  
}  
module.exports.hero = person
```

```

PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL  GITLENS
$ node app.js
{ toBeShared: "I'm to be shared" }

Jeremy@DESKTOP-HPHQ44 MINGW64 ~/Desktop/code/FreeCodeCamp/Backend_And_API/Nodes_Tutorial_1 (main)
$ node modulus.js
Module {
  id: '.',
  path: 'C:\\Users\\Jeremy\\Desktop\\code\\FreeCodeCamp\\Backend_And_API\\Nodes_Tutorial_1',
  exports: {
    items: [ 'item1', 'item2' ],
    hero: { firstname: 'John', lastname: 'Snow' }
  },
  filename: 'C:\\Users\\Jeremy\\Desktop\\code\\FreeCodeCamp\\Backend_And_API\\Nodes_Tutorial_1\\modulus.js',
  loaded: false,
  children: [],
  paths: [
    'C:\\Users\\Jeremy\\Desktop\\code\\FreeCodeCamp\\Backend_And_API\\Nodes_Tutorial_1\\node_modules',
    'C:\\Users\\Jeremy\\Desktop\\code\\FreeCodeCamp\\Backend_And_API\\node_modules',
    'C:\\Users\\Jeremy\\Desktop\\code\\FreeCodeCamp\\node_modules',
    'C:\\Users\\Jeremy\\Desktop\\code\\node_modules',
    'C:\\Users\\Jeremy\\Desktop\\node_modules',
    'C:\\Users\\Jeremy\\node_modules',
    'C:\\Users\\node_modules',
    'C:\\node_modules'
  ]
}

```

Built un Modules

Operating System Module (OS)

OS is a built in module that contains informations about the operating system.

```

const os = require('os');

// Current User inf
const user = os.userInfo();
console.log(user);

```

```

$ node app.js
{
  uid: -1,
  gid: -1,
  username: 'Jeremy',
  homedir: 'C:\\Users\\Jeremy',
  shell: null
}

```

Other OS utilities

```

// Returns the system uptime in seconds
console.log(`The system uptime is ${os.uptime()} seconds`);

const currentOS = {
  name: os.type(),
  release: os.release(),
  totalMem: os.totalmem(),
  freeMem: os.freemem(),
}

```

```
}  
console.log(currentOS);
```

The system uptime is 133809 seconds

```
{  
  name: 'Windows_NT',  
  release: '10.0.19044',  
  totalMem: 16953638912,  
  freeMem: 6310240256  
}
```

Path Module

```
const path = require('path')  
  
// Get the local system path separator  
console.log("My system path separator is : " + path.sep);  
  
// Join is usefull to create pathes that will work on all systems  
const pathfile = path.join('./folder', 'subfolder', 'test.txt')  
console.log(`Here is my path ${pathfile}`);  
  
// Will also make all parts of the path accessible with methods  
const base = path.basename(pathfile);  
console.log(`Here is my file basename ${base}`);  
  
const absolute = path.resolve(__dirname, 'folder', 'subfolder', 'test.txt')  
console.log(`Absolute path is ${absolute}`);
```

```
$ node app.js  
My system path separator is : \  
Here is my path folder\subfolder\test.txt  
Here is my file basename test.txt  
Absolute path is  
C:\Users\Jeremy\Desktop\code\FreeCodeCamp\Backend_And_API\Nodes_Tutorial_1\folder\  
subfolder\test.txt
```