NODE MODULE SYSTEM

NODE MODULE SYSTEM

Every file in a Node application is a module.

Node automatically wraps the code in each file with an IIFE (Immediately-invoked Function Expression) to create scope.

So, variables and functions defined in one file are only scoped to that file and not visible to other files unless explicitly exported

NODE MODULE SYSTEM

To export a variable or function from a module, you need to add them to module.exports

- logger.js

```
var url = 'http://mylogger.com/log'

function log(message){
    console.log(message)
}

module.exports.log = log
```

To load a module, use the require function. This function returns the module.exports object exported from the target module

- app.js

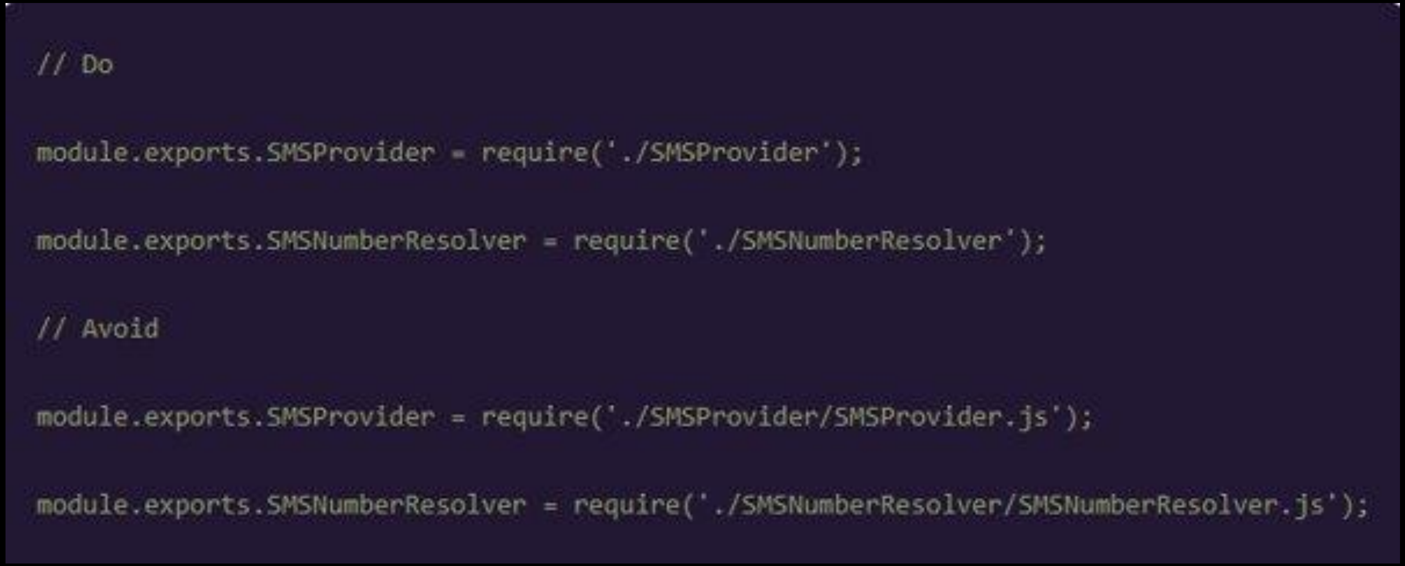
```
const logger = require('./logger.js');

logger.log('message');
```

Require modules first, not inside functions

Require modules at the beginning of each file, before and outside of any functions. This simple best practice will not only help you easily and quickly tell the dependencies of a file right at the top but also avoids a couple of potential problems.

Require modules by folders, as opposed to the files directly

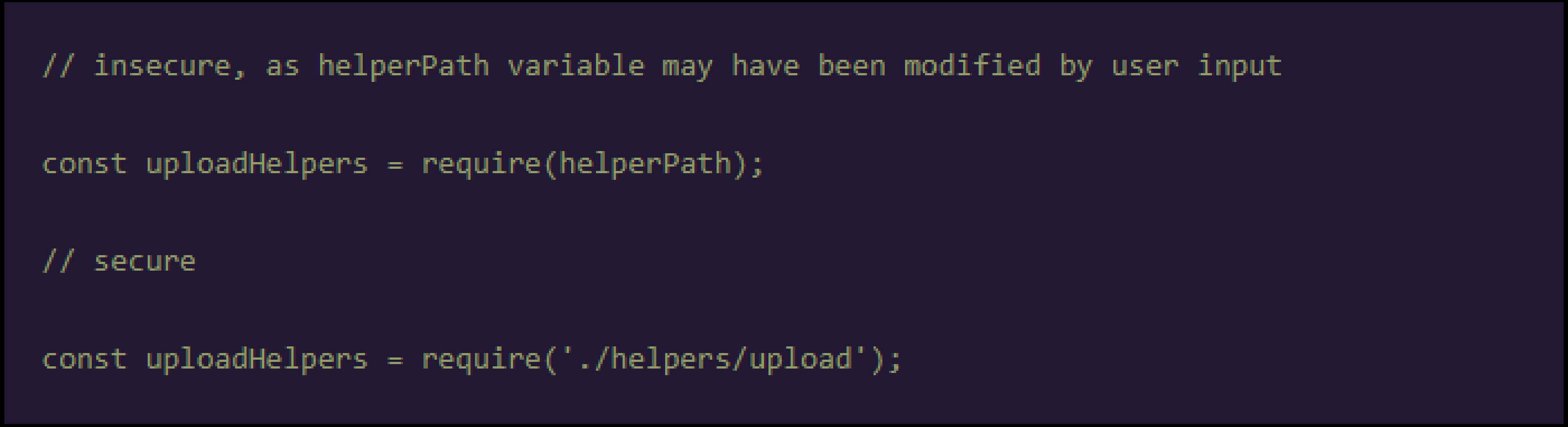
When developing a module/library in a folder, place an index.js file that exposes the module’s internals so every consumer will pass through it. This serves as an ‘interface’ to your module and eases future changes without breaking the contract.



```
// Do

module.exports.SMSProvider = require('./SMSProvider');

module.exports.SMSNumberResolver = require('./SMSNumberResolver');

// Avoid

module.exports.SMSProvider = require('./SMSProvider/SMSProvider.js');

module.exports.SMSNumberResolver = require('./SMSNumberResolver/SMSNumberResolver.js');
```

Avoid module loading using a variable

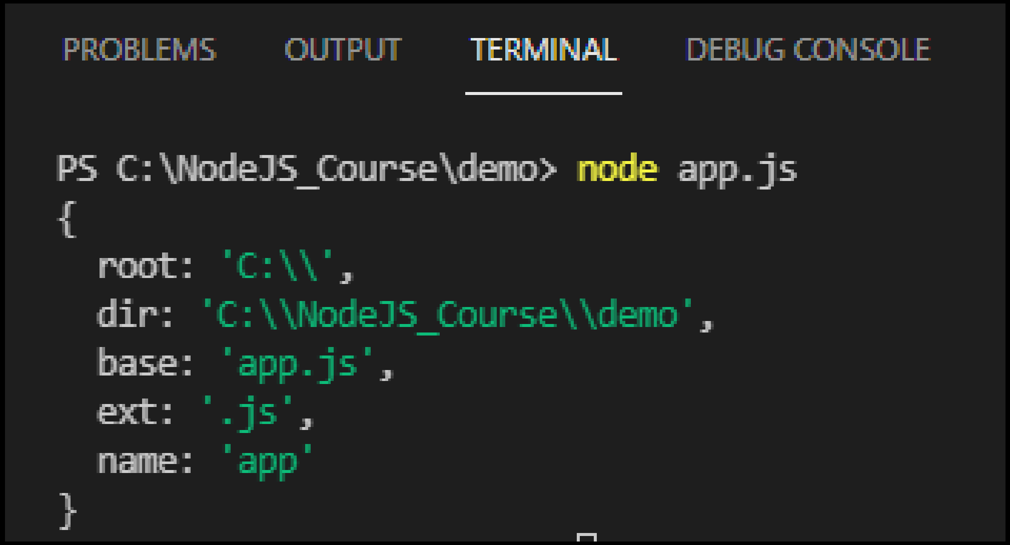
Avoid requiring/importing another file with a path that was given as parameter due to the concern that it could have originated from user input



```
// insecure, as helperPath variable may have been modified by user input

const uploadHelpers = require(helperPath);

// secure

const uploadHelpers = require('./helpers/upload');
```

Path Module

```
const path = require('path');

var pathObj = path.parse(__filename);

console.log(pathObj);
```

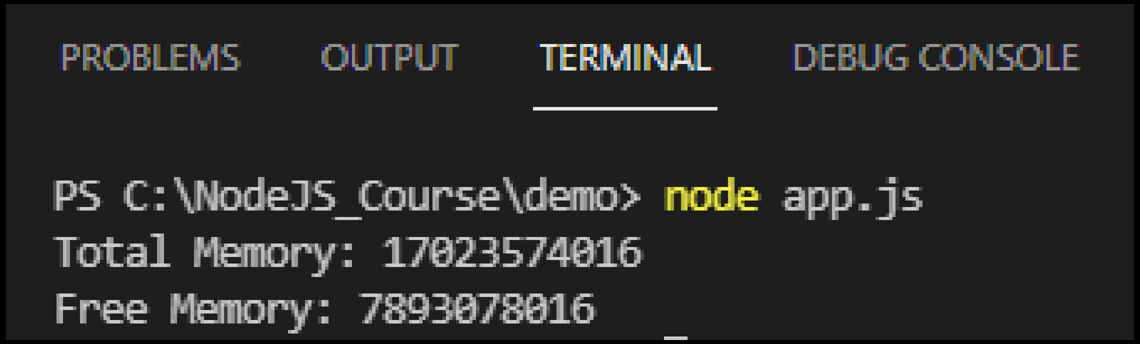


```
PS C:\NodeJS_Course\demo> node app.js
{
  root: 'C:\\',
  dir: 'C:\\NodeJS_Course\\demo',
  base: 'app.js',
  ext: '.js',
  name: 'app'
}
```

NodeJS documentation

OS Module

```
const os = require('os');

var totalMemory = os.totalmem();
var freeMemory = os.freemem();

console.log(`Total Memory: ${totalMemory}`);
console.log(`Free Memory: ${freeMemory}`);
```

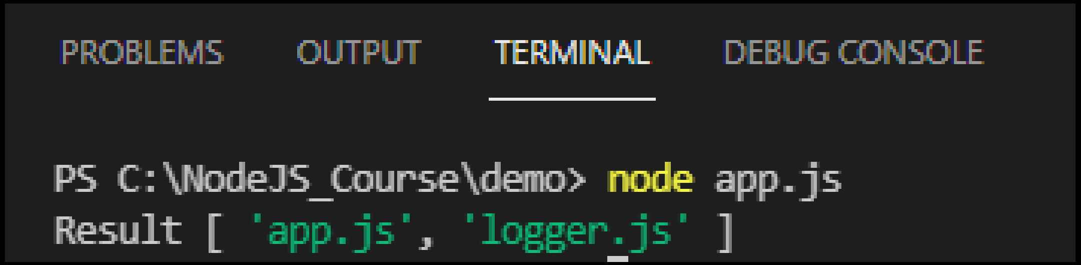


```
PS C:\NodeJS_Course\demo> node app.js
Total Memory: 17023574016
Free Memory: 7893078016
```

NodeJS documentation

File System Module

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

fs.readdir('./', function(err, files){
    if (err) console.log('Error',err);
    else console.log('Result', files);
});
```



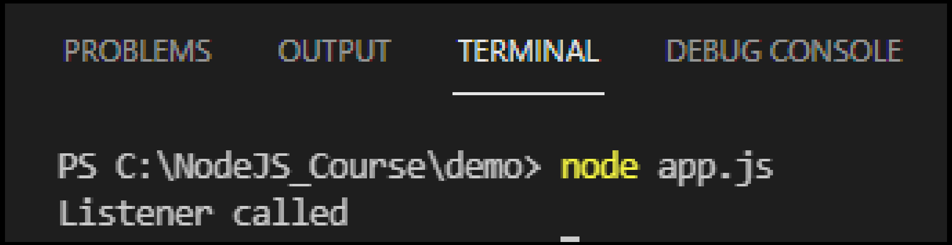
```
PS C:\NodeJS_Course\demo> node app.js
Result [ 'app.js', 'logger.js' ]
```

NodeJS documentation

NODE MODULE SYSTEM

Events Module

```
const EventEmitter = require ('events');
const emitter = new EventEmitter();

//Register a listener
emitter.on('messagerLogged', function(){
    console.log('Listener called')
})

//Raise an event
emitter.emit('messageLogged');
```

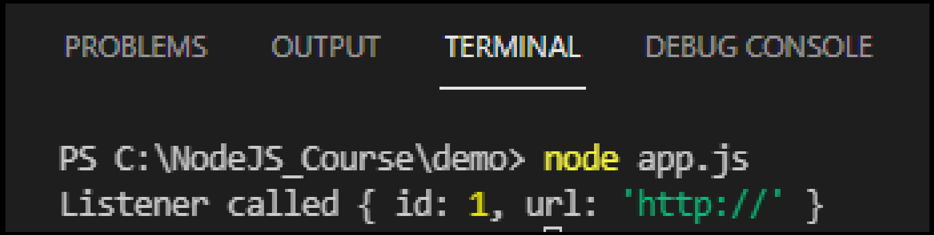


```
PS C:\NodeJS_Course\demo> node app.js
Listener called
```

NodeJS documentation

Event Arguments

```
const EventEmitter = require ('events');
const emitter = new EventEmitter();

//Register a listener
emitter.on('messagerLogged', (arg) => {
    console.log('Listener called', arg)
})

//Raise an event
emitter.emit('messagerLogged', {id: 1, url: 'http://'});
```



```
PS C:\NodeJS_Course\demo> node app.js
Listener called { id: 1, url: 'http://' }
```

NodeJS documentation

Extending EventEmitter

app.js

```
const EventEmitter = require ('events');

const Logger = require('./logger');
const logger = new Logger();

//Register a listener
logger.on('messageLogged', (arg) => {
    console.log('Listener called', arg)
})

logger.log('message');
```

logger.js

```
const EventEmitter = require ('events');

class Logger extends EventEmitter{
    log(message){
        console.log(message)

        //Raise an event
        this.emit('messageLogged', {id: 1, url: 'http://'});
    }
}

module.exports = Logger;
```

NodeJS documentation

EventEmitter

EventEmitter is one of the core classes in Node that allows us to raise (emit) and handle events.

Several built-in classes in Node derive from EventEmitter

NodeJS documentation

HTTP Module

```
const http = require('http');
const server = http.createServer((req, res) =>{
    if (req.url === '/'){
        res.write('Hello World');
        res.end();
    }

    if (req.url === '/api/courses'){
        res.write(JSON.stringify([1, 2, 3]));
        res.end();
    }
});

server.listen(3000);

console.log('Listening on port 3000...');
```

NodeJS documentation