4) Create a Node.js file that writes the file using the File System

Ans.-

Node.js as a File Server

The Node.js file system module allows you to work with the file system on your computer.

To include the File System module, use the require() method:

var fs = require('fs');

Common use for the File System module:

* Read files
* Create files
* Update files
* Delete files
* Rename files

Read Files

The fs.readFile() method is used to read files on your computer.

Assume we have the following HTML file (located in the same folder as Node.js):

demofile1.html

<html>  
<body>  
<h1>My Header</h1>  
<p>My paragraph.</p>  
</body>  
</html>

Create a Node.js file that reads the HTML file, and return the content:

Example[Get your own Node.js Server](https://www.w3schools.com/nodejs/nodejs_server.asp)

var http = require('http');  
var fs = require('fs');  
http.createServer(function (req, res) {  
**fs.readFile('demofile1.html', function(err, data) {**    res.writeHead(200, {'Content-Type': 'text/html'});  
    res.write(data);  
    return res.end();  
  });  
}).listen(8080);

[Run example »](https://www.w3schools.com/nodejs/shownodejs.asp?filename=demo_readfile)

Save the code above in a file called "demo\_readfile.js", and initiate the file:

Initiate demo\_readfile.js:

C:\Users\*Your Name*>node demo\_readfile.js

If you have followed the same steps on your computer, you will see the same result as the example: [http://localhost:8080](http://localhost:8080/)

ADVERTISEMENT

Create Files

The File System module has methods for creating new files:

* fs.appendFile()
* fs.open()
* fs.writeFile()

The fs.appendFile() method appends specified content to a file. If the file does not exist, the file will be created:

Example

Create a new file using the appendFile() method:

var fs = require('fs');  
  
fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {  
  if (err) throw err;  
  console.log('Saved!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_append)

The fs.open() method takes a "flag" as the second argument, if the flag is "w" for "writing", the specified file is opened for writing. If the file does not exist, an empty file is created:

Example

Create a new, empty file using the open() method:

var fs = require('fs');  
  
fs.open('mynewfile2.txt', 'w', function (err, file) {  
  if (err) throw err;  
  console.log('Saved!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_open)

The fs.writeFile() method replaces the specified file and content if it exists. If the file does not exist, a new file, containing the specified content, will be created:

Example

Create a new file using the writeFile() method:

var fs = require('fs');  
  
fs.writeFile('mynewfile3.txt', 'Hello content!', function (err) {  
  if (err) throw err;  
  console.log('Saved!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_writefile)

Update Files

The File System module has methods for updating files:

* fs.appendFile()
* fs.writeFile()

The fs.appendFile() method appends the specified content at the end of the specified file:

Example

Append "This is my text." to the end of the file "mynewfile1.txt":

var fs = require('fs');  
  
fs.appendFile('mynewfile1.txt', ' This is my text.', function (err) {  
  if (err) throw err;  
  console.log('Updated!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_update)

The fs.writeFile() method replaces the specified file and content:

Example

Replace the content of the file "mynewfile3.txt":

var fs = require('fs');  
  
fs.writeFile('mynewfile3.txt', 'This is my text', function (err) {  
  if (err) throw err;  
  console.log('Replaced!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_replace)

Delete Files

To delete a file with the File System module,  use the fs.unlink() method.

The fs.unlink() method deletes the specified file:

Example

Delete "mynewfile2.txt":

var fs = require('fs');  
  
fs.unlink('mynewfile2.txt', function (err) {  
  if (err) throw err;  
  console.log('File deleted!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_unlink)

Rename Files

To rename a file with the File System module,  use the fs.rename() method.

The fs.rename() method renames the specified file:

Example

Rename "mynewfile1.txt" to "myrenamedfile.txt":

var fs = require('fs');  
  
fs.rename('mynewfile1.txt', 'myrenamedfile.txt', function (err) {  
  if (err) throw err;  
  console.log('File Renamed!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_rename)

Upload Files

You can also use Node.js to upload files to your computer.

Read how in our [Node.js Upload Files](https://www.w3schools.com/nodejs/nodejs_uploadfiles.asp) chapter.

Node.js as a File Server

The Node.js file system module allows you to work with the file system on your computer.

To include the File System module, use the require() method:

var fs = require('fs');

Common use for the File System module:

* Read files
* Create files
* Update files
* Delete files
* Rename files

Read Files

The fs.readFile() method is used to read files on your computer.

Assume we have the following HTML file (located in the same folder as Node.js):

demofile1.html

<html>  
<body>  
<h1>My Header</h1>  
<p>My paragraph.</p>  
</body>  
</html>

Create a Node.js file that reads the HTML file, and return the content:

Example[Get your own Node.js Server](https://www.w3schools.com/nodejs/nodejs_server.asp)

var http = require('http');  
var fs = require('fs');  
http.createServer(function (req, res) {  
**fs.readFile('demofile1.html', function(err, data) {**    res.writeHead(200, {'Content-Type': 'text/html'});  
    res.write(data);  
    return res.end();  
  });  
}).listen(8080);

[Run example »](https://www.w3schools.com/nodejs/shownodejs.asp?filename=demo_readfile)

Save the code above in a file called "demo\_readfile.js", and initiate the file:

Initiate demo\_readfile.js:

C:\Users\*Your Name*>node demo\_readfile.js

If you have followed the same steps on your computer, you will see the same result as the example: [http://localhost:8080](http://localhost:8080/)

ADVERTISEMENT

Create Files

The File System module has methods for creating new files:

* fs.appendFile()
* fs.open()
* fs.writeFile()

The fs.appendFile() method appends specified content to a file. If the file does not exist, the file will be created:

Example

Create a new file using the appendFile() method:

var fs = require('fs');  
  
fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {  
  if (err) throw err;  
  console.log('Saved!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_append)

The fs.open() method takes a "flag" as the second argument, if the flag is "w" for "writing", the specified file is opened for writing. If the file does not exist, an empty file is created:

Example

Create a new, empty file using the open() method:

var fs = require('fs');  
  
fs.open('mynewfile2.txt', 'w', function (err, file) {  
  if (err) throw err;  
  console.log('Saved!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_open)

The fs.writeFile() method replaces the specified file and content if it exists. If the file does not exist, a new file, containing the specified content, will be created:

Example

Create a new file using the writeFile() method:

var fs = require('fs');  
  
fs.writeFile('mynewfile3.txt', 'Hello content!', function (err) {  
  if (err) throw err;  
  console.log('Saved!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_writefile)

Update Files

The File System module has methods for updating files:

* fs.appendFile()
* fs.writeFile()

The fs.appendFile() method appends the specified content at the end of the specified file:

Example

Append "This is my text." to the end of the file "mynewfile1.txt":

var fs = require('fs');  
  
fs.appendFile('mynewfile1.txt', ' This is my text.', function (err) {  
  if (err) throw err;  
  console.log('Updated!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_update)

The fs.writeFile() method replaces the specified file and content:

Example

Replace the content of the file "mynewfile3.txt":

var fs = require('fs');  
  
fs.writeFile('mynewfile3.txt', 'This is my text', function (err) {  
  if (err) throw err;  
  console.log('Replaced!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_replace)

Delete Files

To delete a file with the File System module,  use the fs.unlink() method.

The fs.unlink() method deletes the specified file:

Example

Delete "mynewfile2.txt":

var fs = require('fs');  
  
fs.unlink('mynewfile2.txt', function (err) {  
  if (err) throw err;  
  console.log('File deleted!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_unlink)

Rename Files

To rename a file with the File System module,  use the fs.rename() method.

The fs.rename() method renames the specified file:

Example

Rename "mynewfile1.txt" to "myrenamedfile.txt":

var fs = require('fs');  
  
fs.rename('mynewfile1.txt', 'myrenamedfile.txt', function (err) {  
  if (err) throw err;  
  console.log('File Renamed!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_rename)

Upload Files

You can also use Node.js to upload files to your computer.

Read how in our [Node.js Upload Files](https://www.w3schools.com/nodejs/nodejs_uploadfiles.asp) chapter.

Node.js as a File Server

The Node.js file system module allows you to work with the file system on your computer.

To include the File System module, use the require() method:

var fs = require('fs');

Common use for the File System module:

* Read files
* Create files
* Update files
* Delete files
* Rename files

Read Files

The fs.readFile() method is used to read files on your computer.

Assume we have the following HTML file (located in the same folder as Node.js):

demofile1.html

<html>  
<body>  
<h1>My Header</h1>  
<p>My paragraph.</p>  
</body>  
</html>

Create a Node.js file that reads the HTML file, and return the content:

Example[Get your own Node.js Server](https://www.w3schools.com/nodejs/nodejs_server.asp)

var http = require('http');  
var fs = require('fs');  
http.createServer(function (req, res) {  
**fs.readFile('demofile1.html', function(err, data) {**    res.writeHead(200, {'Content-Type': 'text/html'});  
    res.write(data);  
    return res.end();  
  });  
}).listen(8080);

[Run example »](https://www.w3schools.com/nodejs/shownodejs.asp?filename=demo_readfile)

Save the code above in a file called "demo\_readfile.js", and initiate the file:

Initiate demo\_readfile.js:

C:\Users\*Your Name*>node demo\_readfile.js

If you have followed the same steps on your computer, you will see the same result as the example: [http://localhost:8080](http://localhost:8080/)

ADVERTISEMENT

Create Files

The File System module has methods for creating new files:

* fs.appendFile()
* fs.open()
* fs.writeFile()

The fs.appendFile() method appends specified content to a file. If the file does not exist, the file will be created:

Example

Create a new file using the appendFile() method:

var fs = require('fs');  
  
fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {  
  if (err) throw err;  
  console.log('Saved!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_append)

The fs.open() method takes a "flag" as the second argument, if the flag is "w" for "writing", the specified file is opened for writing. If the file does not exist, an empty file is created:

Example

Create a new, empty file using the open() method:

var fs = require('fs');  
  
fs.open('mynewfile2.txt', 'w', function (err, file) {  
  if (err) throw err;  
  console.log('Saved!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_open)

The fs.writeFile() method replaces the specified file and content if it exists. If the file does not exist, a new file, containing the specified content, will be created:

Example

Create a new file using the writeFile() method:

var fs = require('fs');  
  
fs.writeFile('mynewfile3.txt', 'Hello content!', function (err) {  
  if (err) throw err;  
  console.log('Saved!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_writefile)

Update Files

The File System module has methods for updating files:

* fs.appendFile()
* fs.writeFile()

The fs.appendFile() method appends the specified content at the end of the specified file:

Example

Append "This is my text." to the end of the file "mynewfile1.txt":

var fs = require('fs');  
  
fs.appendFile('mynewfile1.txt', ' This is my text.', function (err) {  
  if (err) throw err;  
  console.log('Updated!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_update)

The fs.writeFile() method replaces the specified file and content:

Example

Replace the content of the file "mynewfile3.txt":

var fs = require('fs');  
  
fs.writeFile('mynewfile3.txt', 'This is my text', function (err) {  
  if (err) throw err;  
  console.log('Replaced!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_replace)

Delete Files

To delete a file with the File System module,  use the fs.unlink() method.

The fs.unlink() method deletes the specified file:

Example

Delete "mynewfile2.txt":

var fs = require('fs');  
  
fs.unlink('mynewfile2.txt', function (err) {  
  if (err) throw err;  
  console.log('File deleted!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_unlink)

Rename Files

To rename a file with the File System module,  use the fs.rename() method.

The fs.rename() method renames the specified file:

Example

Rename "mynewfile1.txt" to "myrenamedfile.txt":

var fs = require('fs');  
  
fs.rename('mynewfile1.txt', 'myrenamedfile.txt', function (err) {  
  if (err) throw err;  
  console.log('File Renamed!');  
});

[Run example »](https://www.w3schools.com/nodejs/shownodejs_cmd.asp?filename=demo_fs_rename)

Upload Files

You can also use Node.js to upload files to your computer.

Read how in our [Node.js Upload Files](https://www.w3schools.com/nodejs/nodejs_uploadfiles.asp) chapter.