Lire des fichiers HTML avec Node.js

1. Objectifs

- Être capable d'utiliser **node.js** pour lire un fichier HTML

2. Présentation

- **Node.js** permet de charger des pages ou des applications à partir d'un navigateur grâce à un script serveur simple.
- Dans ce tutoriel nous voyons comment créer un serveur et rendre du HTML dans Node.js.
- Tout d'abord, nous allons créer un nouveau fichier nommé `app.js` dans la racine de notre projet.
- Nous écrivons notre premier serveur dans Node.js! Cela peut sembler étrange si vous venez d'un arrière-plan PHP pour parler de la création de notre premier serveur dans un fichier JavaScript, mais c'est en fait ce que nous sommes sur le point de faire.

3. Lire un fichier HTML

A. Réalisation

1. Créer un serveur sur Node JS

- Pour créer un serveur sur Node JS, nous avons utilisé le module «http». Pour utiliser n'importe quel module dans Node JS, nous devons utiliser le module «require». Nous importons donc d'abord le module «http».
  - `var http = require ( "http" );`

2. Utiliser la méthode 'createServer'

- Maintenant, nous créons un serveur. Pour créer le serveur, nous devons utiliser la méthode '**createServer**' du module http et cette méthode prend 2 paramètres [requête et réponse] comme indiqué ci-dessous.
- Dans l'extrait ci-dessous, le premier ajout que nous apporterons à notre script server.js est d'exiger le module de système de fichiers dans Node.js.

apcpedagogie.com

```
fs.readFile('./index.html',null,(error,data)=>{
        if(error){
            res.end('fill have error')
        }else{
            res.end(data)
        }
    })
```

- **var server = http.createServer (fonction (requête, réponse) {});**

3. **Écouter ce serveur**

- Maintenant, nous devons commencer à écouter ce serveur; sur n'importe quel message http, vous pouvez utiliser n'importe quel port disponible (non utilisé par aucune autre application de votre ordinateur.) Ici, j'ai pris le port 3000.
    - **server.listen (3000);**

B. **Exemples**

1. **Fichier app.js**

- Le paramètre de **createServer** est un callback qui sera activé lorsque le navigateur se connecte au port 1000, avec un nom de fichier en paramètre.

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

const server = http.createServer((req,res)=>{
    fs.readFile('./index.html',null,(error,data)=>{
        if(error){
            res.end('fill have error')
        }else{
            res.end(data)
        }
    })
});
server.listen(3000,'127.0.0.1',()=>{
    console.log('Server running')
})
```

apcpedagogie.com

2. **Fichier index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Bienvenu Je suis votre serveur</h1>
</body>
</html>
```

4. Lire plusieurs fichiers HTML

1. **Exemples**

1. **Fichier app.js**

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

const server = http.createServer((req,res)=>{
    const path=url.parse(req.url).pathname;
    switch(path){
        case '/' :
            afficherPage('./index.html',res)
            break
        case 'contact' :
            afficherPage('./contact.html',res)
            break
        case 'listeCours' :
            afficherPage('./listeCours.html',res)
            break
        case 'user' :
```

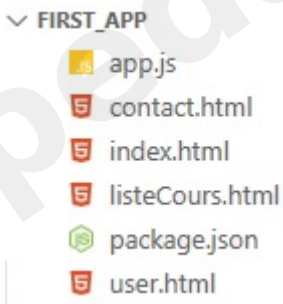
apcpedagogie.com

```
            afficherPage('./user.html',res)
            break
        default:
            res.end('Page non trouvée')
            break
    }

});
server.listen(3000,'127.0.0.1',()=>{
    console.log('Server running')
})
function afficherPage(path,res){
    fs.readFile(path,null,(error,data)=>{
        if(error){
            res.end('Page non trouvée')
        }else{
            res.end(data)
        }
    })

}
```

2. **Fichiers HTML**



Sommaire du cours Node.js

Source : https://apcpedagogie.com/lire-des-fichiers-html-avec-node-js · téléchargé le 04/11/2026 · © APC Pédagogie · apcpedagogie.com