

# Cours de Node.JS

## I- File System :

### 1- Ecrire dans un fichier :

```
1  const data = "Abdilah el gallati Genie informatique"
2  const path = "./data.txt"
3  fs.writeFile(path,data,(err)=>{
4    if(err){
5      console.log(err);
6    }
7    else {
8      console.log("seccess:!!!")
9    }
10 })
```

### 2- Lire un fichier :

```
1  const url = "./data.txt";
2  fs.readFile(url,"utf8",(err,data)=>{
3    if(err) console.log(err) ;
4    else console.log(data);
5  })
```

### 3- Ajouter dans un fichier :

```
1  const source = "  hala hala "
2  const lien = "./data.txt";
3  fs.appendFile(lien,source,(err)=>{
4    if(err) console.log(err);
5    else console.log("succes!!!!");
6  })
```

#### 4- Lire et Ecrire :



```
1  let user={
2    id:12,
3    name:"Mohammed",
4    age:100
5  }
6
7  fs.readFile("./users.json", "utf-8", (err, d)=>{
8    console.log("Reading-----")
9    if(err) return console.log(err)
10   let users=JSON.parse(d)
11   users.push(user)
12   console.log(users)
13   fs.writeFile("./users.json", JSON.stringify(users), (err)=>{
14     console.log(err)
15   })
16 })
17 })
```

## II- HTTP: (HYPER TEXT TRANSFER PROTOCOL)

### 1- HTTP:

Assure la Communication entre un client et un serveur.

### 2- HTTPS (Hyper Text Transfer Protocol Secure):

Echange des données cryptées

Http: méthodes (GET- POST - DELETE - PUT)

## III- Headers :

1XX => Les réponses informative.

2XX => Les réponses de succès

3XX => Les messages de redirection,

4XX => Les erreurs du client

5XX => Les erreurs du serveur

### HTTP: Status Code

- |                           |                               |
|---------------------------|-------------------------------|
| ● 200 : OK                | ● 403 : Forbidden             |
| ● 201 : OK Created        | ● 404 : Not Found             |
| ● 301 : Moved Permanently | ● 405 : Method Not Allowed    |
| ● 304 : Not Modified      | ● 500 : Internal Server Error |
| ● 400 : Bad Request,      | ● 502 : Bad Gateway           |
| ● 401 : Unauthorized      |                               |



```
1  const http=require("http");
2  const fs=require("fs")
3
4  const server=http.createServer((req,res)=>{
5      //console.log(req.url)
6      if(req.url==="/" || req.url=="/index.html"){
7          fs.readFile("index.html","utf-8",(err,data)=>{
8              if(err) {
9                  res.writeHead(500)
10             }
11             else{
12                 res.writeHead(200)
13                 console.log(data)
14                 res.write(data)
15                 res.end()
16             }
17         });
18     }
19     else if(req.url==="/style.css"){
20         res.write('okkkkkkkkkkkkk bn8')
21         res.end();
22     }
23     else{
24         fs.readFile("pnf.html","utf-8",(err,data)=>{
25             if(err) {
26                 res.writeHead(500)
27             }
28             else{
29                 res.writeHead(404)
30                 res.write(data)
31                 res.end()
32             }
33         });
34     }
35 })
36
37 server.listen(3000);
```

## VI- Envoyer différents types de contenu :

- **Content-type** peut être:
  - Du texte brut : **text/plain**
  - Du HTML : **text/html**
  - Du CSS : **text/css**
  - Une image JPEG : **image/jpeg**
  - Une vidéo MPEG4 : **video/mp4**
  - Un fichier ZIP : **application/zip**
  - etc.

```
1 const http = require("http")
2 const url = require("url")
3 const server = http.createServer((req,res)=>{
4   const page = url.parse(req.url).pathname;
5   console.log(page);
6   res.writeHead(200,{"content-type":"text/plain"});
7   if(page=='/'){
8     res.end('Accueil')
9   }
10  else if (page=='/contact'){
11    res.write('Nous contact')
12  }
13  else if (page=='/cours/web/nodejs'){
14    res.write('Node js is great!!')
15  }
16  res.end();
17 });
18 server.listen(3000,()=>{
19   console.log("running.....")
20 })
```

## IV- Gestion des requêtes :

`url.parse(req.url).pathname`

`http://localhost:8080/chemin/vers/la/page`

## V- Gestion des évènements :



```
1  const http = require('http');
2  const server = http.createServer((req,res)=>{
3    res.writeHead(200);
4    res.end("Slut ISIL");
5  })
6  server.on("close",()=>{
7    console.log("Au revoir !!!!")
8  })
9  console.log("Start")
10 server.listen(3000,()=>{
11   console.log("Runnnng.....");
12 })
13 console.log("end")
14 server.close();
```