# Cours de Node.JS

- I- File System:
- 1- Ecrire dans un fichier :

```
const data = "Abdilah el gallati Genie informatique"
const path = "./data.txt"
fs.writeFile(path,data,(err)=>{
   if(err){
      console.log(err);
   }
   else {
      console.log("seccess:!!!")
}
```

2- Lire un fichier:

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

3- Ajouter dans un fichier :

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

#### 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)=>{
6    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 Transferer 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

• 201 : OK Created

• 301 : Moved Permanently

• 304 : Not Modified

• 400 : Bad Request,

• 401 : Unauthorized

403 : Forbidden

404 : Not Found

405 : Method Not Allowed500 : Internal Server Error

• 502 : Bad Gateway

```
• • •
       const http=require("http");
       const server=http.createServer((req,res)=>{
   //console.log(req.url)
   if(req.url==="/" || req.url==="/index.html"){
        fs.readFile("index.html","utf-8",(err,data)=>{
              else if(req.url==="/style.css"){

res.write('okkkkkkkkkkkkk bn8')
                                   res.writeHead(500)
                                   res.writeHead(404)
res.write(data)
       server.listen(3000);
```

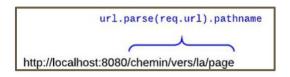
# VI- Envoyer différents types de contenu :

#### • 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.

```
const http = require("http")
const url = require("url")
const server = http.createServer((req,res)=>{
    const page = url.parse(req.url).pathname;
    console.log(page);
    res.writeHead(200,{"content-type":"text/plain"});
    if(page=='/'){
        res.end('Acceuil')
}
else if (page=='/contact'){
        res.write('Nous contact')
}
else if (page=='/cours/web/nodejs'){
        res.write('Node js is great!!')
}
res.end();
}
res.end();
};
server.listen(3000,()=>{
    console.log("runnning.....")
})
```

# IV- Gestion des requêtes :



## V- Gestion des évènements :

```
const http = require('http');
const server = http.createServer((req,res)=>{
    res.writeHead(200);
    res.end("Slut ISIL");
}
server.on("close",()=>{
    console.log("Au revoire !!!!")
}
console.log("Start")
server.listen(3000,()=>{
    console.log("Runnning.....");
}
console.log("end")
server.close();
```