# devweb-tp5

# Partie 1 : Serveur http natif Node.js

## Question 1.1 donner la liste des en-têtes de la réponse HTTP du serveur.

Localhost : Connection keep-alive

Date Fri, 19 Sep 2025 22:35:59 GMT

Keep-alive timeout=5

Transfer-encoding chunked

Favicon.ico : Connection keep-alive

Date Fri, 19 Sep 2025 22 :35 :59 GMT

Keep-alive timeout=5

Transfer-encoding chunked

## Question 1.2 donner la liste des en-têtes qui ont changé depuis la version précédente.

Localhost : connection keep-alive

Content-length 20

Content-type application/json

Date Fri, 19 Sep 2025 22:43:54 GMT

Keep-alive timeout=5

Favicon.ico : connection keep-alive

Content-length 20

Content-type application/json

Date Fri, 19 Sep 2025 22 :43 :54 GMT

Keep-alive timeout=5

## Question 1.3 que contient la réponse reçue par le client ?

Il n’y a pas de réponse par le client car le code appel un fichier index.html qui n’existe pas.

## Question 1.4 quelle est l’erreur affichée dans la console ?

L’erreur affichée dans la console est :

```txt

Error: ENOENT: no such file or directory, open 'C:\UNC\l2\s4\dev\_web\devweb-tp5\index.html'

at async open (node:internal/fs/promises:642:25)

at async Object.readFile (node:internal/fs/promises:1279:14) {

errno: -4058,

code: 'ENOENT',

syscall: 'open',

path: 'C:\\UNC\\l2\\s4\\dev\_web\\devweb-tp5\\index.html'

}

```

## Question 1.5 donner le code de requestListener() modifié avec gestion d’erreur en async/await.

Code modifié:

```js

import fs from "node:fs/promises";

async function requestListener(\_request, response) {

try {

const contents = await fs.readFile("index.html", "utf8");

response.setHeader("Content-Type", "text/html");

response.writeHead(200);

response.end(contents);

} catch (error) {

console.error(error);

response.writeHead(500, { "Content-Type": "text/plain" });

response.end("Erreur 500 : Impossible de charger la page demandee.");

}

}

```

## Question 1.6 indiquer ce que cette commande a modifié dans votre projet.

```js

},

"dependencies": {

"cross-env": "^10.0.0"

},

"devDependencies": {

"nodemon": "^3.1.10"

}

```

## Question 1.7 quelles sont les différences entre les scripts http-dev et http-prod ?

Les différences entre http-dev et http-prod sont :

http-dev démarre le serveur en mode développement avec nodemon, il se relance tout seul quand je change le code et affiche plus de messages pour expliquer ce qu'il se passe.

http-prod démarre le serveur en mode production avec Node.js, il ne se relance pas tout seul et il n’affiche pas les messages de debug.

## Question 1.8 donner les codes HTTP reçus par votre navigateur pour chacune des quatre pages précédentes.