Prérequis

Commençons par rappeler la structure d'une URL[1] :

URL = http://user:pass@sub.example.com:8080/A/B/C/D?query=string#hash

1. URL.path

La propriété path est une concaténation du nom de chemin et des composants de la recherche.

http://user:pass@sub.example.com:8080/A/B/C/D?query=string#hash

ex : '/A/B/C/D?query=string'.

2. URL.pathname

La propriété pathname est constituée de l'intégralité de la section path de l'URL. Il s'agit de tout ce qui suit l'hôte (y compris le port) et avant le début des composants de requête ou de hachage, délimité par les caractères ASCII point d'interrogation (?) ou hachage (#).

http://user:pass@sub.example.com:8080/A/B/C/D?query=string#hash

ex :'/A/B/C/D'.

3. URL.port

La propriété port est la partie numérique du port du composant hôte.

---

[1] https://nodejs.org/api/url.html#url_url_strings_and_url_objects

http://user:pass@sub.example.com:8080/A/B/C/D?query=string#hash

ex :'8080'.

4. URL.protocol

La propriété protocolaire identifie le schéma de protocole en minuscules de l'URL.

http://user:pass@sub.example.com:8080/A/B/C/D?query=string#hash

ex :'http:'.

5. URL.query

La propriété query est soit la chaîne de requête sans le point d'interrogation ASCII (?), soit un objet renvoyé par la méthode parse() du module querystring.

http://user:pass@sub.example.com:8080/A/B/C/D?query=string#hash

ex :'query=string' ou {'query' : 'string"}.

6. URL.search

La propriété de recherche est constituée de la totalité de la partie "chaîne de requête" de l'URL, y compris le point d'interrogation ASCII (?).

http://user:pass@sub.example.com:8080/A/B/C/D?query=string#hash

ex :'?query=string'.

Path

Nous pouvons utiliser le module path pour extraire la variable basename du chemin (la partie finale du chemin) et inverser tout encodage URI du client avec decodeURI comme suit :

const lookup=path.basename(decodeURI(request.url)) ;

decodeURI permet de transformer le codage

Ainsi **decodeURI("https://localhost:3000/A%20%3E%20B/?x=bravo")** donne

`"https://localhost:3000/A > B/?x=bravo"`

Ecrire un fichier server.js

server.js

```
1. const http = require('http');
2. const path = require('path');
3.
4. const pages = [
5.     {route: '', output: 'Woohoo!'},
6.     {route: 'about', output: 'A simple routing with Node example'},
7.     {route: 'another page', output: function() {return `here ${this.route}`}},
8.    ];
9.
10. http.createServer(function (request, response) {
11.    const lookup = path.basename(decodeURI(request.url));
12.    console.log(`lookup: ${lookup}`)
13.    pages.forEach(function (page) {
```

```
14.       if (page.route === lookup) {
15.           response.writeHead(200, { 'Content-Type': 'text/html' });
16.           response.end(typeof page.output === 'function'
17.               ? page.output() : page.output);
18.       }
19.   });
20.   if (!response.writableEnded) {
21.       response.writeHead(404);
22.       response.end('Page Not Found!');
23.   }
24.}).listen(8080);
```

Parsing the querystring

🪛Nous voudrions récupérer l'identifiant par exemple d'un produit dans l'URL.

Par exemple, l'idée est de récupérer la valeur 2 de l'identifiant dans l'adresse localhost:8080/?id=3

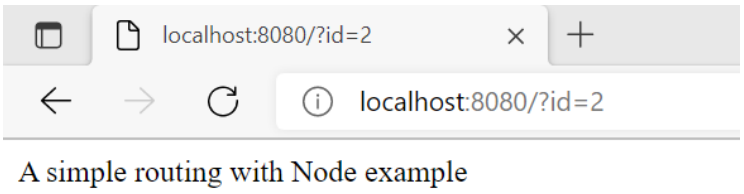
Modifiez le fichier server.js

server.js

```
1. const http = require("http");
2. const url = require("url");
3.
4. const pages = [
5.   { id: "1", route: "", output: "Woohoo!" },
6.   { id: "2", route: "about", output: "A simple routing with Node example" },
7.   {
8.     id: "3",
```

```
    route: "another page",
    output: function () {
      return `here ${this.route}`;
    },
  },
];

http
  .createServer(function (request, response) {
    const num = url.parse(decodeURI(request.url), true).query.id;
    if (num) {
      pages.forEach(function (page) {
        if (num === page.id) {
          response.writeHead(200, { "Content-Type": "text/html" });
          response.end(
            typeof page.output === "function" ? page.output() :
    page.output
          );
        }
      });
    }
    if (!response.writableEnded) {
      response.writeHead(404);
      response.end("Page Not Found");
    }
  })
  .listen(8080, console.log(`listen http://localhost:8080/?id=3`));
```


localhost:8080/?id=2
localhost:8080/?id=2
A simple routing with Node example