

Nodejs : Introduction

Concepts de base

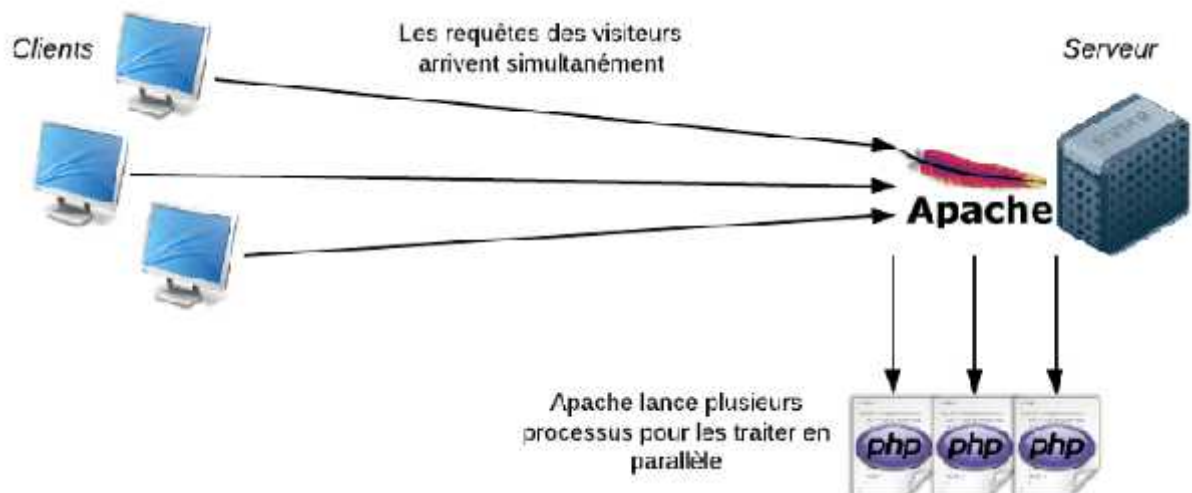
Avant de commencer : du JavaScript côté serveur (exécution du JavaScript côté serveur en dehors du navigateur)

Node.js est un environnement de serveur open source

Node.js est libre

Fonctionnement : *Node.js* vs *Apache*

Apache est Multithread : Lorsque plusieurs visiteurs peuvent demander une page en même temps, Apache peut les exécuter en parallèle en le répartissant sur des différents *threads* (chacun utilise un processeur ou un noyau de processeur). Chaque thread représente un processus PHP par exemple :



Fonctionnement d'Apache [Openclassrooms]

D'abord en Node.js, on n'utilise pas un serveur Web comme Apache ou Nginx. Il faut créer un serveur Web !

Node.js est monothread : un seul processus (une seule version du programme) qui s'exécute en mémoire.

Cependant, Node.js est plus rapide qu'Apache grâce à la programmation asynchrone ou orientée événement. Il utilise moteur d'exécution ultrarapide V8 de Google Chrome et le modèle non bloquant :

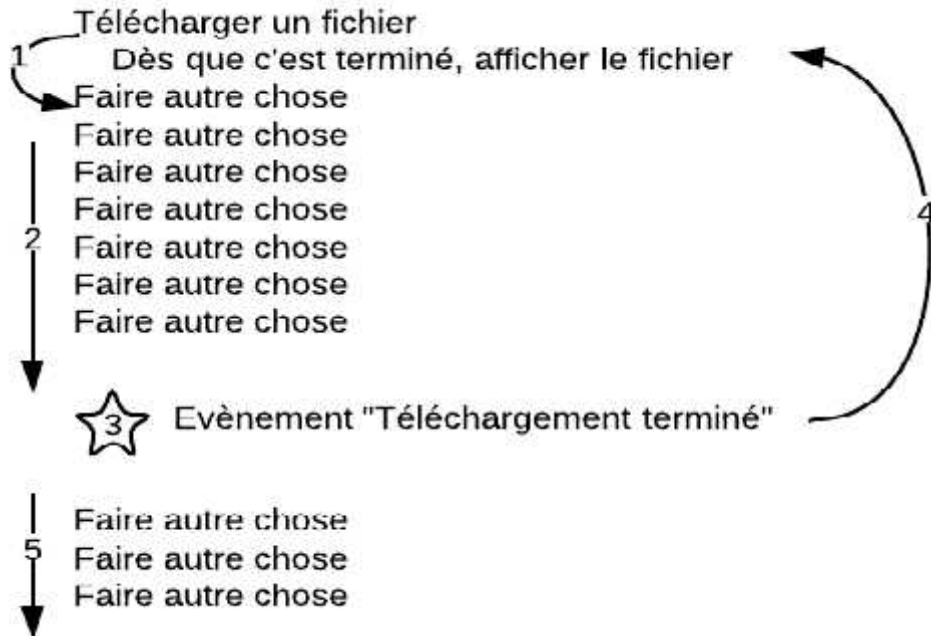
Prenant l'exemple d'une tâche fréquente exécutée par un serveur Web : renvoyer le contenu d'un fichier contenu au client. Un script PHP par exemple gère cette requête come suit :

-) Envoie la tâche au système de fichiers (appel système).
-) Attend que le système de fichiers s'ouvre et lit le fichier.
-) Renvoie le contenu au client.
-) Prêt à gérer la prochaine demande.

Pour Node.js:

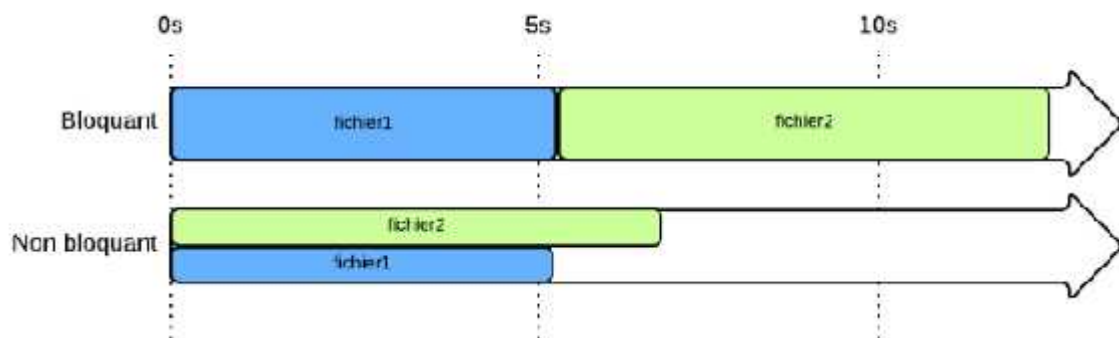
-) Envoie la tâche au système de fichiers.
-) Prêt à gérer la prochaine demande.
-) Lorsque le système de fichiers a ouvert et lu le fichier, le serveur renvoie le contenu au client.

Autrement dit,



Modèle non bloquant avec Node.js [Openclassrooms]

Par exemple, si nous voulons télécharger 2 fichiers, la figure suivante présente la différence entre le mode bloquant et non bloquant :



Modèle bloquant et non bloquant [Openclassrooms]

Exemples de sites et applications qui utilisent Node.js

-) PayPal
-) Netflix
-) Uber
-) LinkedIn mobile app backend
-) Ebay
-) NASA
-) Quelques projets de Mozilla

Installation

Le site web officiel de Node.js contient les instructions d'installation pour Node.js:

<https://nodejs.org>

Premier programme : Hello World

```
var http = require("http");
http.createServer(function(request, response) {
  response.writeHead(200, {"Content-Type": "text/plain"});
  response.write("Hello World");
  response.end();
}).listen(8888);
```

Enregistrer sous server.js

Aller au dossier correspondant=> node server.js

Modules Node.js

-) Les modules sont considérés comme les bibliothèques JavaScript=> Un ensemble de fonctions que vous souhaitez inclure dans votre application.
-) Node.js a un ensemble de modules déjà intégrés => vous pouvez utiliser sans autre installation (comme le module http dans l'exemple précédent). Pour plus d'information : https://www.w3schools.com/nodejs/ref_modules.asp
-) Pour inclure un module, utilisez la fonction require () avec le nom du module:
var http = require("http");

Vous pouvez créer vos propres modules et les inclure dans vos applications. Prenant l'exemple suivant qui représente un module qui renvoie la date et l'heure actuelles:

```
exports.myDateTime = function () {
  return Date();
};
```

Le mot-clé exports rendre les fonctions du module disponibles en externe.

Ensuite, nous pouvons inclure et utiliser le module dans n'importe lequel fichier Node.js. Par exemple si le nom de module est module1.js, nous pouvons l'inclure comme suit:

```
var http = require('http');
var dateTime = require('./module1');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("Current date and time: " + dateTime.CurrentDateTime());
  res.end();
}).listen(8080);
```

"/" signifie que le module est situé dans le même dossier que le fichier Node.js.

Node.js NPM

NPM= Node Package Manager

Il existe des milliers de modules qui offrent des fonctionnalités variées. NPM est un moyen efficace pour installer de nouveaux modules développés par la communauté.

Vous pouvez chercher des modules sur le site web <https://www.npmjs.com/> ou faire une recherche dans la console : **npm search nom_module**

Pour installer un module : **npm install nom_module**

L'installation est locale (pour chaque projet) => cela nous permet de gérer les différences de version

Pour faire une installation globale => **npm install nom_module -g**

Pour mettre à jour les modules => **npm update**

Remarque

Installer ejs et express.js