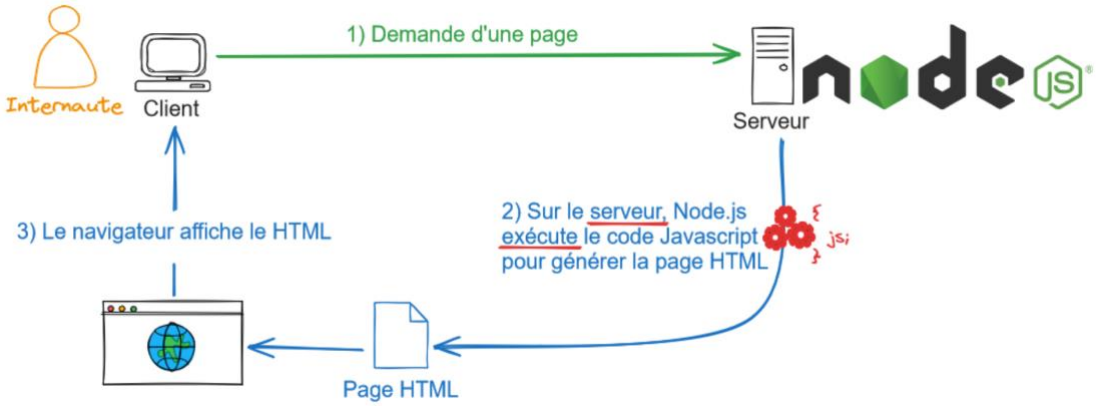
Node.js est une plateforme logicielle open-source qui permet d'exécuter du code JavaScript côté serveur. Avant Node.js, JavaScript était principalement utilisé pour le développement côté client, c'est-à-dire pour les scripts s'exécutant dans le navigateur web de l'utilisateur.

Node.js a permis à JavaScript de s'étendre au développement côté serveur, permettant ainsi aux développeurs d'utiliser un seul langage de programmation (JavaScript) pour le développement côté client et côté serveur.

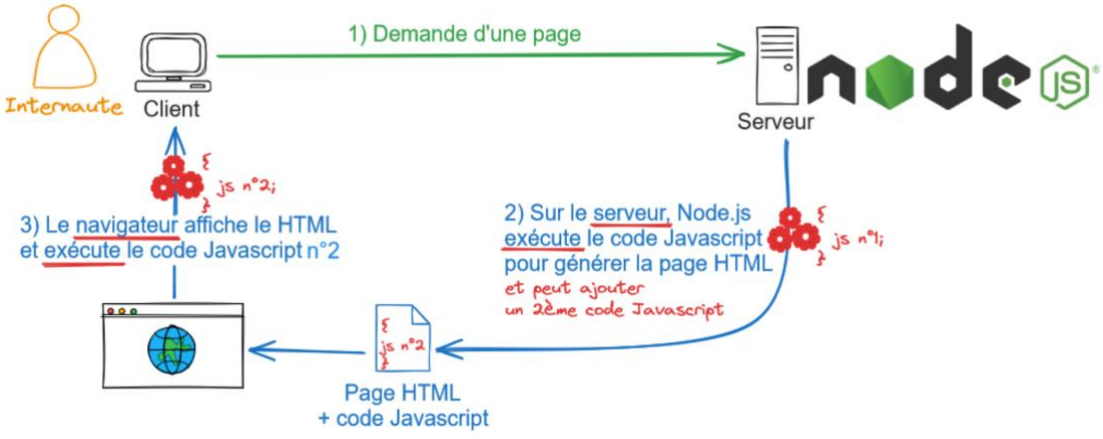
Les points clés

Exécution côté serveur : Node.js permet d'exécuter du JavaScript côté serveur. Cela signifie que vous pouvez utiliser JavaScript pour interagir avec votre système d'exploitation, avec votre base de données, manipuler des fichiers, etc., tout comme vous le feriez avec d'autres langages de programmation côté serveur comme Python, Ruby ou PHP.



Basé sur le moteur V8 de Google : Node.js est construit sur le moteur JavaScript V8 de Google Chrome. V8 est incroyablement rapide et continue de s'améliorer avec le temps. Il compile et exécute le JavaScript à une vitesse impressionnante en le convertissant en code machine.

Asynchrone et non bloquant : Node.js utilise un modèle d'entrées/sorties (I/O) asynchrone et non bloquant. Cela signifie que Node.js peut gérer plusieurs opérations simultanément sans avoir à attendre qu'une opération se termine avant de passer à la suivante. Cela le rend très efficace pour gérer des applications à fort trafic ou pour réaliser plusieurs opérations I/O simultanément.

NPM (Node Package Manager) : Node.js est livré avec un gestionnaire de paquets intégré appelé NPM. NPM est utilisé pour installer et gérer des paquets de code qui peuvent être utilisés dans vos applications Node.js. Il existe des milliers de paquets disponibles qui peuvent vous aider à accomplir une variété de tâches, allant de la manipulation d'images à la gestion de bases de données.

Utilisé pour le développement back-end et full-stack : Node.js est couramment utilisé pour le développement d'applications back-end, notamment les API RESTful et les serveurs web. De plus, avec l'émergence de frameworks comme Express.js pour le back-end et Angular, React ou Vue pour le front-end, Node.js est également une option populaire pour le développement full-stack.