

Formation Node.js avancée

Objectifs de la formation Node.JS

Node.js est un projet open-source reposant sur le moteur V8 de Chrome. Il permet d'exécuter du code JavaScript côté serveur, contrairement à ce que l'on a l'habitude de voir avec le JavaScript côté client. Node.js est un interpréteur JavaScript exécutable et enrichissant le langage avec sa propre API. Sa spécificité vient de son API : entièrement orientée vers le non bloquant, elle permet d'écrire des applications avec d'excellents temps de réponse. Ce cours Node.js vous permettra de maîtriser les concepts avancés de Node.js, tant d'un point de vue objets avancés JavaScript, programmation fonctionnelle, paradigme asynchrone, ES6, que de performance et qualité des applications Node.js.

Concrètement, à la fin de ces 3 jours de formation, les participants seront en mesure de :

- Maîtriser le cœur de la technologie Node.js
- Optimiser les performances de leurs applications
- Améliorer la qualité des applications.

À qui s'adresse cette formation ?

Public :

Ce cours s'adresse aux développeurs, aux architectes et aux chefs de projets.

Prérequis :

Pour suivre ce cours, il est nécessaire de maîtriser JavaScript et d'avoir une première expérience de Node.js.

Contenu du cours Node.JS

Rappels sur Node.js

Installation et utilisation du REPL : présentation nvm&co JavaScript côté serveur : démonstrations et premiers travaux pratiques La "single-threadedeventloop" API non bloquante : intérêts Aller plus loin avec les objets JavaScript : Object.create, Object.defineProperty La programmation fonctionnelle : map, reduce, currying (illustration avec lodash)

La programmation asynchrone

Dompter le paradigme asynchrone Les différentes API : callback et librairies associées, fibers, promesses Les avantages et pièges à éviter Gérer la soupe de callbacks avec Async

Les modules Node.js

Description Fonctionnement Structure

Écrire un serveur avec Node.js

Présentation Démonstration

Les librairies d'accès aux bases de données

Présentation NoSQL : MongoDB et Redis Modélisation avec NoSQL Différentes utilisations de Redis Implémentation du modèle

Communication en temps réel

Définition et problématiques Les technologies à disposition Intégration des WebSockets HTML5 avec Socket.IO

Communication inter-process en temps réel

Le pub/sub avec Redis AMQP dans Node JS Présentation RabbitMQ et ZeroMQ Event-loop distribuée

Les tests avec NodeJS

Tests unitaires avec Mocha (atelier) Tests fonctionnels avec les headless browsers Intégration avec npm

Travaux Pratiques :

Exercices sur les server-sideevents, websockets avec socket.io

Gestion de la performance avec JavaScript et Node.js

Écrire du JavaScript performant pour V8 : les bonnes pratiques Gestion de la mémoire : la pile et la mémoire totale, comment les gérer Anticiper et trouver les fuites mémoire Effectuer des calculs lourds : pool de workers, amqp... Les Cluster Utiliser tous les processeurs de sa machine Cluster et données partagées La solution haute performance Redis Bonus spécial troll Comparaison avec l'équivalent Apache/PHP

Qualité

Déboguer son application : utilisation du débogueur v8 Profiling : état des lieux, outils Bonnes pratiques : les meilleurs outils de test, contrôle de qualité ducode, conventions... Discussion libre

Conclusion et conseils

Synthèse des bonnes pratiques de développement Node.js Veille : comment faire le tri dans les milliers de modules disponibles ? Le futur avec ES6 : let, const, arrows, classes, destructuration, proxy, observation, generators, comprehensivelists, collections, promesses... Le futur présent : fonctionnalités ES6 déjà utilisables dans Node

Travaux Pratiques

Ce stage est une alternance de présentation magistrale (30%) et de live coding. Le stage suit pour fil conducteur la réalisation d'une application simple et fonctionnelle, permettant la mise en application concrète des différents concepts abordés.