

Programme de formation Node.js

Découvrez la puissance du Javascript coté serveur avec cette formation Node.js

Durée

3 jours / 21 heures

Description

Node.js est devenu aujourd'hui incontournable, "npm" est le nouveau "apt-get" pour les développeurs web, et quand on parle de websockets le nom "socket.io" revient immanquablement dans la discussion.

Ce succès n'est évidemment pas immérité et cette plateforme, basée sur le runtime JavaScript V8 de Chrome, permet d'obtenir des performances époustouflantes, tout en renouant avec un plaisir de programmer qu'on a parfois perdu de vue avec les technos habituelles...

Nous pourrons ensemble découvrir ce qui fait de Node.js une plateforme unique, agréable, et tellement productive.

Objectifs pédagogiques

- Comprendre le fonctionnement de Node.js, ses forces et ses faiblesses.
- Être capable de créer une application Node.js complète, en suivant les bonnes pratiques de développement.
- Être capable de choisir les bons modules npm pour la bonne tâche.
- Être capable de détecter les situations où Node.js sera un choix adapté et le justifier

Pré-requis

- Connaissances de base du développement web (HTML, CSS, JavaScript)
- Connaissances des principes de développement côté serveur (la formation n'est pas adaptée aux développeurs uniquement frontend)
- Installation de XCode pour ceux qui sont sous Mac
- Ordinateur portable à apporter

Méthodes pédagogiques

40 % théorie / 60 % pratique

Profil intervenant

L'ensemble de nos formations sont animées par des formateurs expérimentés possédant une expérience terrain éprouvée.

Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers d'ateliers de mise en pratique des notions et concepts abordés pendant la formation.

Programme

Jour 1 Découverte

Présentation de Node.js

- JS côté serveur, pourquoi et comment ?
- Présentation de Node.js (REPL, single-threaded event-loop, APIs non bloquantes, streams...)
- Programmation asynchrone : joies & désillusions (Continuous Passing Style, Promise, Fibers, bonnes pratiques de gestion de ses appels asynchrones)
- Les modules et npm

Anatomie d'un module

- La gestion de dépendances avec npm, le gestionnaire de paquets officiel de Node.js
- Packaging et déploiement d'un module sur npm

Tests

- Tests unitaires avec Mocha, le framework de tests unitaires, et Chai, la librairie d'assertions intuitive
- Tests fonctionnels avec Zombie, Phantom et Casper, les navigateurs "headless"

Application web avec Node.js

- Écriture d'un serveur web
- Utilisation du module "Express", le micro-framework web le plus populaire (routage, vue, middlewares, montage d'applications)
- Gestion des fichiers statiques et comparaison avec les serveurs existants

Jour 2 Les mains dans le cambouis

Notre application

- Présentation du projet développé durant le reste des travaux pratiques
- Discussions autour des technologies à utiliser et du modèle de données

Node.js et les bases de données

- SQL (MySQL, PostgreSQL, SQLite...)
- NoSQL (MongoDB, Redis, et bonnes pratiques)
- Modélisation

Enhance your webapp

- Aller plus loin avec Express (traitement de formulaires et authentification avec passport, le gestionnaire de sessions polyglote)
- Websockets avec Socket.io (présentation, intégration côté serveur et côté client, et nombreux ateliers)

Jour 3 – Vers l’infini et au-delà !

Déléguer les traitements lourds

- Les limites imposées par le modèle single-thread de Node.js
- Les solutions pour déléguer ces traitements à d’autres processus (pub/sub Redis, AMQP, ØMQ, Gearman... mais aussi le multi-threading)

Aller plus loin dans les performances

- Comment rendre son application "scalable"
- Profiter de tous les cœurs de son processeur avec cluster
- Trollons joyeusement avec un comparatif de performances

Conclusions & conseils

- Synthèse des bonnes pratiques de développement
- Conseils de survie dans l’écosystème Node.js