



Node.js

Durée 3 jour(s) (NODEJS-03)

Maîtrisez la puissance de JavaScript côté serveur

Description

Node.js est une plate-forme Web permettant de développer des applications asynchrones côté serveur en langage JavaScript, offrant des nouvelles possibilités permettant de gérer un grand volume d'utilisateurs.

Objectifs

- Connaître l'architecture et le fonctionnement de Node.js
- Savoir installer et configurer Node.js et son écosystème de plugins
- Comprendre les concepts d'une architecture asynchrone
- Savoir créer des applications Node.js
- Mettre en oeuvre les meilleures pratiques de développement et d'utilisation de Node.js

Public

- Développeur
- Architecte Logicielle

Prérequis

- Connaître et maîtriser les fondamentaux JavaScript
- Une première expérience en développement Serveur

Répartition

40% Théorie, 60% Pratique

Programme

Rappel des bonnes pratiques JavaScript

- Visibilité des variables
- Structuration de code
- Les Closures
- Objets et prototypes
- Lodash

Introduction à Node.js

- Origine du projet
- Le moteur d'interprétation Chrome V8
- La notion de gestion d'événements
- Les différents usages
- Overview de l'écosystème de plugins
- Installation
- Un tout premier exemple

Architecture de Node.js

- Asynchronisme
- Programmation par callbacks
- L'event loop

Modules et gestion de dépendances

- L'approche modulaire
- NPM et manipulation des modules
- Le fichier package.json en détail
- Modularisation de son code
- Modules noyaux: console, process, os, fs, path et util
- Publier un module sur NPM

Node et le Web : HTTP, Connect & Express

- Périmètre du module HTTP
- Connect et ses middlewares
- Serveur Web avec Express
- Générer un serveur Express generator
- Router les requêtes
- Gestion d'une requête et création d'une réponse
- Utiliser un moteur de template
- Ouverture à Passport et Hapi

L'asynchrone en détails

- Callback Hell & Pyramid of Doom
- Le module async
- Les promesses
- Exemples d'enchaînement et de parallélisation
- Gestion des erreurs

Communication temps réel

- Intégration des WebSockets HTML5
- Socket.io
- Gestion des transports
- Intégration côté client et côté serveur

La gestion des streams

- Description de Streams2
- Buffers
- Type de flux
- Pipelining
- Object mode
- Les outils : through2, trumpet, JSONStream...
- Gulp

Liaison avec la persistance des données

- Approche via le driver ou un ORM
- Intégration avec une base de données relationnelle
- Modules d'abstraction SQL
- Le module Sequelize
- Intégration avec une base de données NoSQL
- Le module Mongoose

Outillage et Usine Logicielle

- Outils d'édition
- Utilisation des logs
- Outils de débogage
- Tests unitaires avec Mocha
- Modules Chai et Sinon
- Nodemon
- L'orchestration des tâches (build) avec Grunt
- Intégration dans le serveur d'intégration Jenkins

Node.js en mode Cluster

- Utilisation du module cluster
- Master et workers