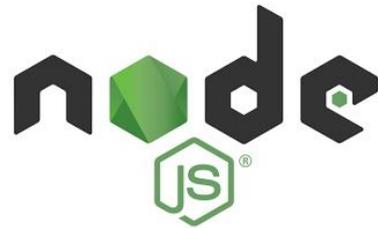


Formation



Formation Node.js

-

Programmation JavaScript côté serveur

CodeWise

formations pour développeur web

Formation Node.js

Présentation du projet

Intitulé de la formation :

Formation Node.js, programmation JavaScript côté serveur

Présentation :

Cette formation vous apprendra à développer des applications réactives et performantes avec Node.js. Vous utiliserez les API de Node.js et implémenterez la persistance de vos données dans une base NoSQL.

Cela vous permettra de mettre en œuvre les concepts de programmation événementielle, de programmation asynchrone, de modularité et de routage.

Format/Durée : 28h sur 4 jours.

La formation se fait en présentiel ou à distance, avec un suivi individuel et un examen d'évaluation en fin de formation.

Environnement de travail :

Suivi par un formateur-expert à toutes les étapes de la formation. Bureaux en centre-ville de Toulouse pour les parties présentiels. Matériel informatique fourni.

Audience : Développeurs/architectes, avec des bases Javascript, CSS et HTML

Tarif : 2490€ HT

Objectifs pédagogiques :

- Installer et configurer un serveur Node.js
- Mettre en œuvre les concepts de la programmation événementielle et asynchrone
- Mettre en place un framework Web
- Manipuler l'API de Node.js
- Gérer la persistance dans une base de données NoSQL avec un ODM

Programme de la formation

Du JavaScript côté serveur

- Genèse du JavaScript côté serveur et de Node.js.
- Le moteur Google V8 utilisé côté serveur.
- Pourquoi utiliser la programmation événementielle ?
- Rappels JavaScript : callbacks, closures, notion de scope, apply, bind, call.
- Initiation à ECMAScript 6.

Premiers pas en Node.js

- Installation du serveur Node.js.
- Le gestionnaire d'extensions NPM.
- L'approche modulaire de Node.js
- Un serveur Web en quelques lignes.
- Utilisation de Node.js en REPL.

Travaux pratiques

Usage de l'utilitaire NPM. Développer une première application.

Les fondamentaux Node.js

- Quel intérêt de développer en asynchrone ?
- La gestion événementielle : réagir plutôt qu'attendre.
- Principaux modules de l'API : console, util, file, events & timer...
- Gestion des requêtes/réponses HTTP. HTTPS.
- Création de processus fils, https, sockets TCP et UDP...

Travaux pratiques

Lecture de fichier, lecture d'une ressource en ligne, création d'un processus fils avec communication IPC.

La gestion de routes

- Parsing d'URL (paramètres, requête...).
- Traiter une requête en mode asynchrone.
- Mettre en place un gestionnaire de routes.

Travaux pratiques

Mise en place d'un gestionnaire de routes.

Framework Web

- Les concepts fondamentaux d'Express.
- Construction d'un squelette d'application.
- Configuration d'Express et de l'application.
- Le rendu de vues avec EJS.
- La gestion de formulaires et des uploads de fichiers.
- Le routage d'URL par Express.
- Mise en place d'une API REST.

Travaux pratiques

Mise en place d'une API REST complète.

Persistence des données

- Initiation à une base NoSQL : MongoDB.
- Mise en place de CRUD (Create, Read, Update, Delete).
- Utilisation d'un ODM : Mongoose.

Travaux pratiques

Gestion d'une persistance au travers d'un ODM avec création des modèles.

Test d'une application Node.js

- Premiers pas avec Mocha.
- Les assertions et le test synchrone et asynchrone.
- Différentes méthodes de tests (inclusifs, exclusifs...).

Travaux pratiques

Tester une application avec Mocha.

Concept avancés

- Bonnes pratiques de développement en Node.js.
- La bibliothèque async : rendre l'asynchrone séquentiel.
- Le clustering avec Node.js. La retro-compatibilité, les transpilers...

