

# Node.js

## Description

Durant ce cours de 5 jours, les participants apprendront à concevoir, tester et déployer une API RESTful avec Node.js et Mongo. La formation offre une approche pratique, centrée sur la création d'applications robustes et sécurisées adaptées aux environnements professionnels. Chaque étape est guidée pour permettre aux apprenants de construire un projet complet, de la configuration initiale jusqu'au déploiement.

## Pourquoi suivre une formation Node.js

La formation Node.js couvre l'essentiel des technologies modernes utilisées par les développeurs. Les participants découvrent le rôle des middlewares, l'organisation d'une application et la modélisation des données avec Mongoose. Ils apprennent aussi à intégrer des tests, sécuriser les routes grâce aux JSON Web Tokens et gérer l'authentification. Enfin, la formation se termine par des étapes concrètes de déploiement afin de rendre le projet opérationnel en production.

### Contenu du cours

#### Module 1 : Programme d'installation

- Introduction
- Installation de Node
- Qu'est-ce que Node?
- Pourquoi devrais-je utiliser Node?

#### Module 2 : Notions de base de Node.js

- Utiliser Require
- Require de vos propres fichiers
- Utiliser des modules tiers
- Redémarrage de l'application avec Nodemon
- Obtenir des saisies de l'utilisateur
- Entrée simplifiée avec Yargs
- Travailler avec JSON
- Refactoring pour la réutilisabilité
- Notes de lecture et réutilisabilité
- Débogage des applications Node.js
- Débogage via Chrome Dev Tools
- Identifier les Notes
- Exiger des arguments et des arguments avancés
- Fonctions de flèche (arrow)

#### Module 3 : Node.js asynchrone

- Notions de base asynchrones
- Pile d'appel et boucle d'événement
- Fonctions de callback et API
- Objets d'impression

- De quoi est faite une requête HTTP?
- Codage de l'entrée de l'utilisateur
- Erreurs de callback
- Résumé des rappels
- Câblage de la météo
- Chaînage des rappels ensemble
- Intro aux promesses ES6 (Promises)
- Promesses avancées
- Fonctionnalités supplémentaires

#### **Module 4 : Serveurs Web et déploiement d'applications**

- Créer un serveur Web
- Rendu de modèles avec des données
- Templates avancés
- Express Middleware
- Ajouter un contrôle de version (Git)
- Configuration des clés GitHub et SSH
- Déployer vos applications
- Ajouter une nouvelle fonctionnalité et déployer

#### **Module 5 : Test de vos applications**

- Mocha et tests de base
- Tests de surveillance et de redémarrage automatique
- Utilisation d'une bibliothèque d'assertion
- Test du code asynchrone
- Test des applications Express: Partie I
- Test des applications Express: Partie II
- Test d'organisation avec describe()

#### **Module 6 : API MongoDB, Mongoose et REST**

- Installation de MongoDB et Robomongo (Windows)
- Construire un vocabulaire NoSQL
- Connexion à Mongo et écriture de données
- L'ObjectId
- Récupération des données
- Mise en place du Repo
- Supprimer des documents
- Mise à jour des données
- L'ORM Mongoose
- Mise en place de Mongoose
- Validateurs, types et valeurs par défaut
- Installation de Postman
- Point de fin de la création de ressources - POST / todos
- Requêtes Mongoose et validation d'ID
- Déployer l'API sur Heroku

#### **Module 7 : Sécurité et authentification**

- Configuration du modèle utilisateur

- JWT et Hashing
- Génération de jetons Auth et définition des en-têtes
- Routes privées et middleware Auth
- Hashing mots de passe
- Seeding Base de données de test avec les utilisateurs
- Amélioration de la configuration de l'application

## **Module 8 : Applications Web en temps réel avec Socket.io**

- Créer un nouveau projet
- Ajouter Socket.io à une application
- Émettre et écouter des événements
- Événements de broadcasting
- Générateur de messages et tests
- Reconnaissance des événements
- Formulaire de message et jQuery
- Géolocalisation Partie I
- Géolocalisation Partie II
- Timestamps et formatage avec Moment
- Impression des Timestamps des messages
- Moustache.js
- Autoscrolling
- Ajouter une page
- Données de passage
- Socket.io Rooms
- Stockage des utilisateurs avec des classes ES6: Partie I
- Stockage des utilisateurs avec des classes ES6: Partie II
- Liste des utilisateurs
- Envoyer des messages seulement à la Room
- Nouvelles idées de fonctionnalités

## **Module 9 : Async / Await**

- Async / Await Project Setup
- Async / Await Basics
- Un exemple du monde réel
- Gestion des erreurs et attente de la fonction asynchrone
- Utilisation d'Async / Await dans l'API Todo

## **Lab / Exercices**

- Pendant le cours, les participants sont encouragés à participer activement à l'expérience d'apprentissage en exécutant des exemples de fichiers et en effectuant des tâches de codage pendant les travaux de lab. Chaque session de lab vous permet de comparer votre solution à celle de l'instructeur

## **Documentation**

- Support de cours numérique inclus

## **Profils des participants**

- Développeurs web souhaitant élargir leurs compétences
- Futurs freelances en développement Node.js

- Administrateurs système voulant automatiser des services
- Étudiants ou reconvertis en programmation web

### **Connaissances Préalables**

- Bonne compréhension de base JavaScript standard (variables, instructions if, fonctions de base, objets de base)

### **Objectifs**

- Construire et déployer une API entièrement fonctionnelle pour une application avec authentification
- Créer leurs propres API
- Prétendre à un poste de développeur Node professionnel capable de développer, tester et déployer des applications de production dans le monde réel

### **Description**

Formation Node.Js

### **Niveau**

Fondamental

### **Prix de l'inscription en Présentiel (CHF)**

3800

### **Prix de l'inscription en Virtuel (CHF)**

3550

### **Durée (Nombre de Jours)**

5

### **Reference**

NODEJS