

## Node.js JavaScript : développer des applications web réactives

Date et durée
Code formation : APINODE Durée : 5 jours Nombre d'heures : 35 heures
Description
<p>Le JavaScript, oublié par un grand nombre de développeurs, est aujourd'hui considéré comme <b>un véritable moteur de création d'API</b>. Pour preuve, il suffit de regarder les statistiques qui indiquent clairement que sur le portail GitHub, les dépôts JavaScript sont plus nombreux que les projets de développement en JS ou même en Python.</p> <p>Certes, si ce retour est en grande partie dû au récent moteur d'exécution JavaScript développé par Google pour Chrome, il est également lié aux bibliothèques qui sont venues enrichir les fonctionnalités du langage. Parmi ces librairies, le Framework Node.js offre la liberté de <b>concevoir du JavaScript sans passer par un navigateur</b>. De ce fait, côté serveur, les temps de traitement et de chargement sont plus rapides.</p> <p>Dans cette formation, vous apprendrez les concepts de base de la programmation JavaScript ainsi que les concepts avancés du <b>développement web avec node.js</b> et le framework Express. Vous serez en mesure de concevoir des API en maîtrisant les principes de routage et de programmation événementielle asynchrone. L'interaction entre <b>le front-end et le back-end</b> sera bien détaillée avec des illustrations pour les bases de données relationnelles et NoSQL. Par ailleurs, vous participerez tout au long des 12 cours proposés à de nombreux <b>travaux pratiques</b>.</p>
Objectifs
<p>Cette <b>formation Node.js</b> vous permet de valider les objectifs de compétences suivants :</p> <ul style="list-style-type: none"><li>• installer et configurer un serveur web Node.js et ses composants ;</li><li>• maîtriser la programmation événementielle et asynchrone ;</li><li>• installer et paramétrer un Framework Web ;</li><li>• concevoir, optimiser et sécuriser une API ;</li><li>• gérer la couche de persistance avec des données structurées et non structurées.</li></ul>
Points forts
<p>Une formation avec de nombreux travaux pratiques à chaque cours, un formateur expert dans le développement Java et Node.js, une réelle montée en compétences et le passage d'un questionnaire d'auto-évaluation.</p>
Modalités d'évaluation
Travaux Pratiques
Pré-requis
Suivre la <b>formation Node.js</b> nécessite les prérequis suivants :

- avoir des connaissances avancées en JavaScript et connaître le langage HTML et CSS.

Public

**Cette formation s'adresse aux publics suivants :**

- les développeurs d'applications, les architectes API et les chefs de projets techniques.

Cette formation s'adresse aux profils suivants

Développeur

Architecte logiciel / Applicatif

Ingénieur logiciel

Chef de projet / Responsable de projet

Programme

### **Cours 1 : généralités sur le langage de programmation JavaScript**

- La syntaxe de base du langage JavaScript.
- L'histoire de JavaScript et de Node.js côté serveur.
- L'usage des fonctions de rappel en JavaScript (callbacks, closures, scope, apply, bind, call).
- Présentation de la version 6 d'ECMAScript.

### **Cours 2 : premier pas avec Node.js**

- Qu'est-ce qu'un serveur Web en node.js ?
- L'installation du serveur web Node.
- Le gestionnaire d'extensions NPM.
- L'approche modulaire de Node.js
- L'utilisation de Node.js dans REPL.

### **Cours 3 : les bases de Node.js**

- La programmation fonctionnelle et le mode asynchrone.
- La gestion événementielle : réagir plutôt qu'attendre.
- Les principaux modules de l'API (console, util, file, events, timer, etc.).
- La gestion des requêtes et des réponses (HTTP et HTTPS).

### **Cours 4 : les bases d'Express.js**

- Qu'est ce que l'intergiciel (middleware) ?
- La création de sa première application avec ExpressJs
- Le routage.
- Le traitement des demandes.
- Les vues.

### **Cours 5 : l'utilisation du middleware ExpressJs**

- Les différents type de middleware.
- La gestion des erreurs.

### **Cours 6 : l'utilisation du routage ExpressJs**

- Le concept de routage.
- Les caractéristiques du routage (gestion des paramètres).
- L'utilisation des expressions régulières.
- Le routage (conception d'applications).
- Le routage et l'optimisation de la gestion des contenus web statiques.

## **Cours 7 : la conception des APIs**

- Les API JSON.
- Le parsing d'une URL (paramètres, requête, etc.).
- Le traitement d'une requête en mode asynchrone.
- L'API CRUD
- La gestion des erreurs.

## **Cours 8 : les vues et les modèles en Jade et EJS**

- Les vues d'Express.
- La modélisation avec le moteur de template EJS.
- La syntaxe Jade.
- Les modélisations avec le moteur de template Jade.

## **Cours 9 : la persistance des données**

- Le concept de NoSQL (MongoDB).
- La bibliothèque Mongoose.
- L'implémentation de CRUD (Create, Read, Update, et Delete).
- Le pilote de nœud MongoDB.
- L'utilisation de Mysql et TypeORM.

## **Cours 10 : les tests d'une application Node/ExpressJs**

- L'utilisation de Mocha, supertest et Jasmine.
- Les assertions dans les tests synchrones et asynchrones.
- Les différentes méthodes de test.

## **Cours 11 : les Framework web basés sur la technologie Nodejs**

- Présentation de Hapi.dev.
- Présentation de Sails.js.
- Présentation de Nest.js.
- Présentation de Restify.js.

## **Cours 12 : le déploiement d'applications avec Heroku**

- L'optimisation du code et des actifs (LESS, Browserify, Grunt et Minify).
- Les étapes de déploiement avec Heroku.
- L'utilisation de Grunt sur Heroku.