

# NodeJS

## INFORMATIONS GÉNÉRALES

**Type de formation :** Formation continue

**Éligible au CPF :** Non

**Domaine :** Développement

**Action collective :** Non

**Filière :** Java JEE

**Rubrique :** Java/JEE

**Code de formation :** LEDN214

## PRÉSENTATION

### Objectifs & compétences

A l'issue de la formation, le stagiaire sera capable :

- Maîtriser les fonctionnalités principales de NodeJS
- Savoir utiliser NodeJS, NPM et son écosystème dans les dernières versions
- Savoir configurer un serveur de NodeJS
- Développer une application web avec NodeJS et ES2022
- Maîtriser la programmation événementielle & asynchrone
- Créer et gérer APIs avec NodeJS
- Sécuriser, industrialiser, tester & déployer son application

### Public visé

Développeurs web

### Pré-requis

Connaissance du langage JavaScript  
Connaissance d'un framework côté client ou d'un autre langage de programmation orienté objet (java, php, etc.)

## € Tarifs

**Prix public :** 2430 €

### Tarif & financement :

Nous vous accompagnons pour trouver la meilleure solution de financement parmi les suivantes :

**Le plan de développement des compétences de votre entreprise :** rapprochez-vous de votre service RH.

**Le dispositif FNE-Formation.**

**L'OPCO** (opérateurs de compétences) de votre entreprise.

**France Travail:** sous réserve de l'acceptation de votre dossier par votre conseiller Pôle Emploi.

**CPF -MonCompteFormation**

Contactez nous pour plus d'information : [contact@aston-institut.com](mailto:contact@aston-institut.com)

## 📍 Lieux & Horaires

**Durée :** 28 heures

### Délai d'accès :

**8 jours avant le début de la formation**

## PROGRAMME

### Jour 1 – Introduction aux principes fondamentaux

#### Rappel de JavaScript

- L'histoire du langage
- Les principes fondamentaux du langage
- La boucle d'événement ou l'Event Loop
- Les moteurs JavaScript
- Focus sur le moteur V8 de Google

#### ES2022

- Initiation à ECMAScript
- Déclaration de variables et portée
- Littéraux objets
- Le format JSON
- Les classes
- Déstructuration
- Rest et Spread
- Template strings
- Les fonctions fléchées
- Les modules ES
- Compatibilité native Node
- Utiliser la dernière version de JavaScript grâce à Babel

## 📅 Prochaines sessions

Consultez-nous pour les prochaines sessions.

## Programmation asynchrone

- Les callbacks
- Les callbacks selon NodeJs
- Le problème du "callback hell"
- Utiliser async.js pour éviter le callback hell
- Les promesses
- Async control flow avec async / await

## Introduction à NodeJs

- La genèse de NodeJs
- Exécuter du JavaScript côté serveur
- Installation du serveur Nodejs
- Un premier programme
- Exécuter un fichier
- Présentation globale de l'API de Node.Js
- Comparaison avec d'autres technologies

## Présentation des principaux composants Node.JS

- Node CLI (outils en ligne de commande)
- Les différents environnements de développement (IDE)
- NPM – Le gestionnaire de paquets de node.js
- package.json
- Les Node Modules
- Outils : Development Tools et Frameworks

## Les objets globaux

- Focus sur la documentation de l'API de Node
- L'objet global et différence avec window
- Utilisation des fonctions setTimeout, setInterval et setImmediate
- logging sur process.stdout avec console
- Accès au contexte du fichier avec \_\_dirname et \_\_filename
- Accès à la configuration hardware du server process et os

## Jour 2 – Manipulation de l'API de Node

### Gestion des modules Node

- Qu'est-ce qu'un module Node ?
- Les modules core
- Import de module avec require et import
- Configuration de module et initialisation de module
- Utilisations des modules utilitaires (util, path, queryString, url)
- Création de modules

### Découverte de NPM

- Le gestionnaire de paquet
- L'outil en ligne de commande npm
- L'alternative yarn
- Recherche de module en ligne de commande
- Le site npmjs.com
- Recherche de module sur le site
- Installation locale ou global
- Packaging de module
- Le fichier package.json
- Déclaration des dépendances
- Gestion des conflits de version
- Gestion de dépendances par environnement

### Manipulation de fichier

- Présentation du module fs
- Lecture de fichier synchrone
- Lecture de fichier asynchrone
- Création de fichiers asynchrone
- Suppression de dossier asynchrone

### Programmation événementielle

- Pourquoi la programmation événementielle
- Présentation du module events
- Utilisation de EventEmitter
- Exemple d'utilisation concret

### **Jour 3 – Développement d'application web**

#### **Accès aux réseaux depuis NodeJS**

- Rappel de réseau
- Les modules core Node orienté réseaux
- Utilisation des module udp et net
- Utilisation des module http et http2
- Utilisation du module dns
- Zoom sur le protocole HTTP

#### **Création d'un serveur web avec l'api Node.JS**

- Qu'est-ce qu'un serveur HTTP ?
- Lancement d'un serveur web Node
- Gestion des requêtes/réponses HTTP
- Mise en place d'un gestionnaire de routes
- Traitement de requête de manière asynchrone

#### **Création d'un serveur web avec Express**

- Introduction à Express
- Comparaison avec Fastify et NestJS
- Lancement d'un serveur express
- Configuration d'une application Express avec les middlewares
- Utilisation du gestionnaire de routes d'Express
- Les moteurs de templating
- Création de template Pug et rendering d'une page HTML
- Traitement de formulaire HTML

#### **Connexion à une base de données**

- Les bases de données compatibles
- Introduction à MongoDB
- Utilisation du package mongoose : création de modèle et requêtage
- Lier une route à un modèle mongoose
- Restifier un modèle de données avec express-restify-mongoose
- Utilisation du package sequelize : création de modèle et requêtage
- Lier une route à un modèle sequelize
- Restifier un modèle de données avec finale-rest

#### **Communication bidirectionnelle temps réel**

- Introduction à Websocket
- Présentation de socket.io
- Gestion de la communication côté serveur
- Gestion de la communication côté client

### **Jour 4 – Industrialisation d'une application Node.JS**

#### **Builder votre projet**

- Pourquoi builder un projet nodeJs ?
- Les outils de build
- Rédiger ses propres scripts
- Partir d'un projet boilerplate (style Yeoman)

#### **Tester et déboguer**

- Les modules Node core pour tester et déboguer (console, debugger, inspector, repl, assert)
- L'écosystème des packages npm orienté testing (unitaire et intégration)
- Modules d'assertion : assert et Chai
- Tester son module avec Mocha

#### **L'écosystème des packages NPM**

- Bien choisir un package npm: analyse de viabilité
- Les principaux frameworks de développement d'API
- Qui sont les développeurs de package npm ?
- Comment contribuer à un package npm ?

**Sécurisation d'une application Node/Express**

- Les modules core de sécurité (crypto, https, tls)
- Encryption de mot de passe avec bcrypt
- Le package helmet
- Authentification avec Passeport

**Faciliter le développement d'application Node en équipe**

- Versionner proprement votre code avec git
- Documentation du code avec docco
- Documentation d'une API à l'aide de Swagger
- Harmonisation d'une base de code à l'aide de ESLint
- Imposer le typage via Typescript ou Flow

**MODALITÉS****Modalités**

Jusqu'à 8 jours avant le début de la formation, sous condition d'un dossier d'inscription complet

**Méthode**

**Fin de formation :** entretien individuel.

**Satisfaction des participants :** questionnaire de satisfaction réalisé en fin de formation.

**Assiduité :** certificat de réalisation.

**Validations des acquis :** grille d'évaluation des acquis établie par le formateur en fin de formation.