

Formation NodeJS

Durée :	4 jours
Public :	Développeurs Javascript
Pré-requis :	Bonne connaissance de javascript et des normes ecma-script
Objectifs :	Apprendre à utiliser NodeJS et créer un serveur
Sanction :	Attestation de fin de stage mentionnant le résultat des acquis
Taux de retour à l'emploi:	Aucune donnée disponible
Référence:	SIT100450-F
Note de satisfaction des participants:	4,93 / 5

Introduction

- Présentation des versions de node
- Principe d'un serveur Web
- Comparaison entre serveur
- Rappels de Javascript

Atelier : Télécharger et installer NodeJS

Présentation de NodeJS

- Gérer les versions de node (nvm vs n)
- L'intérêt de javascript côté serveur
- La single-threaded event loop
- L'API non bloquante
- Le moteur V8 de Google

NodeJS et Ecma-script

- Historique d'Ecma-script
- Présentation du mode strict
- Support d'ecma-script dans NodeJS (node.green)

Utiliser le gestionnaire de paquets

- Présentation et description
- npm: node package manager
- Initialiser un projet avec npm
- Présentation de la bibliothèque de module
- Installation locale ou globale

Le fichier package.json
Gérer les dépendances (production et développement)
npm et npx
Ajouter des scripts

Atelier : Créer différents scripts utiles à un projet

Les variables globales

La documentation de l'API de Node
Les classes Global, Console (log, errors, trace)
Les variables __filename et __dirname
Capter et émettre des événements
Les Timers

Atelier : Executer notre premier script

Manipuler les fichiers

Présentation du module FileSystem
Gérer les chemins et les urls
Créer, supprimer, renommer des fichiers et dossiers
Lire le contenu d'un fichier

Atelier : Manipuler des fichiers

Callback et Promise

Présentation des callbacks (fonction de rappel)
Principe des Promises (les promesses)
Callbacks vs Promises
Implémentation des promesses
Transformation d'une fonction de rappel en promesse
Travailler les promesses avec la syntaxe async/await

Atelier : Utiliser les promesses

Créer des modules NodeJS

Isoler des fonctionnalités
Exporter des fonctions
Inclure des modules dans un projet
Différence entre module CommonJS et EcmaScript

Atelier : Créer ses propres modules

Créer un serveur avec NodeJs

Le module "http"
Request et Response
Modifier les entêtes
Parser une url
Afficher des contenus au format HTML, JSON, XML
Gérer l'affichage des fichiers css, image et js

Atelier : Créer un serveur de fichiers statiques

Créer un serveur avec Express

- Présentation de "Express"
- Créer un projet Express
- Les middlewares
- Gérer les verbs http
- Ajouter un moteur de template
- Récupérer les paramètres de requête, d'url et de formulaire
- Les modules Morgan, Helmet, Cors
- Utiliser Express avec Typescript

Atelier : Utiliser Express pour créer une application Web ou une API REST avec authentification

Les bases de données

- Se connecter à des bases de données SQL et NoSQL
- Récupérer un connecteur de SGBD
- Utiliser et configurer un ORM: TypeORM
- Créer une classe d'entité
- Executer des requêtes et récupérer les resultats
- Mettre en place des relations entre deux entités

Atelier : Connecter une base de données

Tester une application

- Présentation de Mocha
- Les assertions dans NodeJS: assert
- Le module Chai
- Les tests synchrones et asynchrones
- Les hooks

Atelier : Lancer plusieurs test