

# Node.js

Des bases solides pour développer des applications back-end en JavaScript

3jours / 21h

## Public cible

- Développeurs, intégrateurs, architectes logiciel, chefs de projet technique

## Programme

- Avoir suivi le cours JavaScript: Les fondamentaux du langage ou avoir des connaissances équivalentes

## Programme

1. Introduction à Node.js
  - Qu'est-ce que Node.js ?
  - Les outils: node et npm
  - Environnement de développement Node.js
  - Notions d'architecture Node.js: IO, asynchrone, non bloquant, concurrence, event-loop, scalability
  - Ateliers:
    - Installation de Node.js et d'un environnement de développement
    - Démonstration de la event loop à travers un premier programme asynchrone / non-bloquant
    - REPL

## 2. Évènements

- Architecture de Node.js basée sur les évènements
- Event et EventEmitter
- Synchrones vs Asynchrones
- Gestion des évènements multiples ou unique
- Gestion des erreurs
- Ateliers:
  - Développement d'EventEmitter personnalisé

## 3. Streams

- Exemples de données streaming gérées par Node.js
- Les différents types de Streams
- Buffers internes
- Pipelines
- Ateliers:
  - Consommer une stream Readable (http, fs)
  - Ecrire dans une stream Writable (fs)
  - Exploiter une stream Duplex (net.Socket)
  - Utiliser une stream Transform (zlib)

## 4. Gestion d'erreurs

- Les différents types d'erreurs
- Gestion d'erreur dans les APIs synchrones
- Gestion d'erreur dans les APIs asynchrones
- Assertions
- Logging via l'API Console
- Debugger un programme Node.js
- Ateliers:
  - Gestion d'erreur dans les APIs synchrones
  - Gestion d'erreur dans les APIs asynchrones
  - Débugger un programme Node.js dans son IDE ou dans Chrome

## 5. Modularité

- Object Global et scope inter-modules
- Objects accessibles globalement
- Modules, scopes et gestion des fichiers
- Core modules
- Module main
- Algorithme de chargement des modules et cache de modules
- NPM et modules tiers
- Ateliers:
  - Utilisation de l'objet Global

- Objets accessibles globalement: process
- Usage de core modules: timer, OS, Util, Path, ...
- Création de module interne à l'application
- Création d'un projet Node.js via NPM et installation de modules populaires

## 6. Clients/Serveurs TCP et UDP

- Module Net pour création de serveurs et clients TCP
- Module UDP/Datagram pour création serveurs et clients UDP/Datagram
- Ateliers:
  - Création de clients / serveur TCP
  - Création de clients / serveur UDP/Datagram

## 7. Clients/Serveurs http, https et http/2

- Modules http, https et http/2 pour la création de serveurs et clients http, https et http/2
- Module url pour exploiter les requêtes http
- Ateliers:
  - Création de serveurs et clients http, https et http/2 dans le cadre d'un site web
  - Création de serveurs et clients http, https et http/2 dans le cadre d'un webservice RESTFul

## 8. Interactions avec le système de fichiers

- Le module fs et sa proximité avec POSIX
- APIs synchrones vs APIs asynchrones
- Ateliers:
  - Interactions avec le système de fichiers via APIs synchrones et asynchrones

## 9. Développement de back-end web

- Gestion des données avec bases de données Relationnelles (Mysql) ou NoSQL (Mongo, Redis)
- Gestion du middleware (connect, expressjs)
- Gestion de templates (ejs)
- Ateliers:
  - Création d'un back web statique et dynamique avec express.js
  - Création d'une API RESTFul avec express.js

## 10. Déployer une application Node.js

- NPM: les principales commandes
- NPM: packager un module
- Différents modes de déploiement d'un projet Node.js
- Modules Forever et PM2
- Ateliers:

- Packager une librairie
- Dockerizer une application web Node.js