

Formation

Node.js & API REST

Formation présentielle ou en ligne à la demande. Apprenez à concevoir, développer et sécuriser des API REST professionnelles avec Node.js et Express, en appliquant les bonnes pratiques de développement backend.

Formation 100% pratique

Exercices & cas d'usage

Formateurs expérimentés

4 jours

35 h

Format

Atelier
pratique

Modalité

Présentiel
ou en ligne

Niveau

Intermédiaire

INSCRIPTION / RÉSERVATION



Je m'inscris
maintenant



OBJECTIFS PÉDAGOGIQUES

- Comprendre l'architecture et le modèle d'exécution de Node.js.
- Construire des API REST robustes avec Express et les normes HTTP.
- Gérer la validation des données, les erreurs, l'authentification et la sécurité.
- Mettre en production une API Node.js (logging, tests, déploiement, bonnes pratiques).

PUBLIC CIBLE

- Développeurs JavaScript souhaitant passer au backend avec Node.js.
- Développeurs Fullstack souhaitant standardiser leurs APIs REST.
- Équipes techniques impliquées dans des projets web & mobiles.

PRÉREQUIS

- Bonne maîtrise des bases de JavaScript (ES6 : fonctions, objets, promesses).
- Notions de base sur HTTP, JSON et les architectures web.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Introduction à Node.js & rappel JavaScript backend

Comprendre Node.js

- Historique, cas d'usage typiques de Node.js.
- Moteur V8, single-thread, event loop, non-blocking I/O.
- Environnement d'exécution Node vs navigateur.

Rappel JavaScript moderne (ES6+)

- let/const, fonctions fléchées, template strings.
- Destructuring, spread/rest operator.
- Promesses, async/await, gestion basique des erreurs.

Ecosystème Node.js : modules, npm & structuration du projet

Travailler avec npm & modules

- Installation de Node.js, npm, nvm.
- Initialisation de projet, package.json, scripts npm.
- Modules CommonJS & ES Modules, require/import.
- Gestion des dépendances : dependencies vs devDependencies.

Organisation et bonnes pratiques

- Arborescence type d'une API Node.js.
- Variables d'environnement (.env) et configuration.
- Utilisation de nodemon, scripts de démarrage.

Fondamentaux des API REST & mise en œuvre avec Express

Concevoir une API REST

- Principes REST, ressources, verbes HTTP, codes de statut.
- Structuration des URL, versioning, conventions de nommage.
- JSON comme format d'échange – bonnes pratiques.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Créer une API avec Express

- Installation et configuration de Express.
- Routes, middleware, handlers, next().
- Implémenter les opérations CRUD (Create, Read, Update, Delete).
- Gestion des paramètres, query string, body (JSON).

Persistance des données & intégration avec une base Connexion à une base de données

- Choix de la base : SQL vs NoSQL (ex. PostgreSQL, MongoDB).
- Connecteurs / ORM (ex. Sequelize, Mongoose – selon le cas choisi).
- Gestion des connexions, pooling, erreurs de base.

Modélisation & accès aux données

- Modèles, schémas, relations simples.
- Implémentation des opérations CRUD persistantes.
- Gestion des identifiants, clés primaires, pagination simple.

Validation, gestion des erreurs, logging & sécurité

Validation & gestion des erreurs

- Validation des données d'entrée (ex. Joi, Zod, class-validator).
- Middleware de validation centralisé.
- Gestion des erreurs applicatives et techniques.
- Format standard de réponse d'erreur (codes & payload JSON).

Sécurité d'une API REST

- Principes OWASP pour les APIs.
- CORS, rate limiting, protection contre les attaques courantes.
- Stockage sécurisé des secrets et tokens.

Logging & monitoring

- Logs structurés (ex. Winston, pino).
- Corrélation des requêtes, niveaux de log, traces.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Authentification, autorisation & gestion des utilisateurs

Modèle de sécurité applicative

- Concepts d'authentification vs autorisation.
- Stateless vs stateful, sessions vs tokens.

JWT & rôles

- Création d'un flux d'authentification (login, refresh token).
- Protection des routes par middleware.
- Gestion des rôles et permissions (admin, user, etc.).

Intégration avec un client

- Consommation de l'API depuis un front (SPA / mobile – présentation).

Qualité, tests, documentation & déploiement

Tests & qualité de code

- Tests unitaires & d'intégration (ex. Jest, supertest).
- Stratégie de tests pour une API REST.
- Linting, formatage (ESLint, Prettier).

Documentation de l'API

- Documentation avec OpenAPI/Swagger.
- Exposer une UI de test (Swagger UI / ReDoc).

Déploiement & exploitation

- Build & configuration pour l'environnement de prod.
- Déploiement sur un PaaS ou VM (ex. Azure, Docker, etc.).
- Introduction à PM2, gestion des redémarrages & logs en production.

Atelier fil rouge : développer une API REST complète

Projet de bout en bout

- Définition d'un besoin métier et des ressources API.
- Conception du modèle de données et des endpoints.
- Implémentation des routes, validation, persistance et sécurité.
- Tests, documentation, préparation au déploiement.