

Formation MERN, Développement Full Stack (Mongo, Express, React, Node)

Présentation

Cette formation avancée MERN vise à concevoir et industrialiser une application full stack professionnelle, performante et sécurisée. Sur cinq jours, les participants optimiseront l'architecture (API, front, data), mettront en œuvre des fonctionnalités critiques (auth, validation, gestion d'état, cache) et évalueront la qualité via tests, observabilité et pratiques DevOps.

Un fil rouge encadré permet de livrer une application complète, prête pour un déploiement en environnement entreprise.

Durée : 35,00 heures (5 jours)

Tarif INTRA : Nous consulter

Objectifs de la formation

- Concevoir une architecture MERN modulaire et scalable (API, front, data, conventions et packaging)
- Optimiser la conception des données MongoDB (schémas, index, agrégations) et évaluer les performances
- Concevoir et sécuriser une API Node/Express (auth, autorisations, validation, sécurité applicative)
- Optimiser une application React professionnelle (routing, état, performance, accessibilité et UX)
- Mettre en œuvre une stratégie de tests et de qualité (unitaires, intégration, E2E, CI) et évaluer la couverture
- Concevoir un pipeline de livraison et d'exploitation (build, déploiement, observabilité, gestion de configuration)

Prérequis

- Maîtrise de JavaScript moderne (ES2015+) et notions TypeScript souhaitables
- Expérience en Node.js et en consommation/écriture d'API REST
- Connaissance des bases React (hooks, composants, state/props)
- Notions de base sur une base de données (requêtes, modèles de données)
- Pratique de Git et d'un workflow de développement (branches, PR)



Public

- Développeurs full stack confirmés, leads techniques et ingénieurs logiciels souhaitant maîtriser une stack MERN en contexte produit.
- Contextes : applications web métier, portails à fort trafic, MVP industrialisés, modernisation d'applications et intégrations SI.
- Expérience attendue : pratique régulière de JavaScript/TypeScript, Node.js et React, avec compréhension des APIs HTTP et des bases de données.

Programme de la formation

Jour 1 – Architecture MERN pro, outillage et socle du projet

Session du matin :

- Concevoir l'architecture : découpage front/API, contrats, conventions, monorepo vs polyrepo et structure de dossiers
- Outillage pro : TypeScript, ESLint/Prettier, Git hooks, standards de commit et gestion de configuration
- Design d'API : ressources, pagination/filtrage, erreurs standardisées et versioning

Session de l'après-midi :

- Mise en place Express : routing, middleware, validation d'entrées, gestion centralisée des erreurs
- Connexion MongoDB : configuration, environnements (dev/test/prod), seeds et scripts
- Front React : bootstrap projet, structure, routing de base et composants de layout

TP / Exercice :

- Démarrer le fil rouge : initialiser un workspace (front + API), configurer TypeScript, lint/format, scripts et variables d'environnement. Créer une première ressource API (CRUD minimal) et une page React de listing connectée. Livrable : repository prêt (scripts, README, conventions) et socle exécutable.

Points clés & takeaways :

- Structurer une base de code MERN industrialisable (conventions, scripts, configuration)
- Mettre en place un socle API + front cohérent et exécutable en environnement local
- Définir des contrats d'API stables et des erreurs standardisées

Jour 2 – MongoDB avancé : modélisation, indexation et performance**Session du matin :**

- Modéliser les données : embedded vs referenced, relations, contraintes, validations et stratégie d'identifiants
- Indexation : types d'index, cardinalité, index composés, unique, TTL et impact sur les écritures
- Requêtes et agrégations : pipelines, projections, regroupements et optimisation (explain)

Session de l'après-midi :

- Mongoose pro : schémas, hooks, transactions (notion), pagination, soft delete et patterns repository
- Optimiser les accès : limitation des champs, tri, pagination robuste, prévention N+1 côté API
- Mesurer : métriques de latence, analyse d'index, identification des goulots d'étranglement

TP / Exercice :

- Faire évoluer le modèle fil rouge : définir schémas et relations, ajouter index pertinents, implémenter pagination/filtrage côté API et produire une agrégation (ex. statistiques). Mesurer et améliorer une requête lente avec explain et ajustements d'index. Livrable : schémas versionnés + endpoints optimisés + note de performance avant/après.

Points clés & takeaways :

- Choisir un modèle MongoDB adapté et le faire évoluer sans casser l'API
- Optimiser requêtes et agrégations via indexation et lecture ciblée
- Évaluer l'impact performance et documenter des choix de modélisation

Jour 3 – API sécurisée : authentification, autorisations et sécurité applicative

Session du matin :

- Sécuriser l'API : menaces courantes, validation serveur, sanitation et politique d'erreurs
- AuthN/AuthZ : JWT, refresh token, sessions, RBAC/ABAC et principes de moindre privilège
- Sécurité Express : CORS, headers, rate limiting, protection brute force et gestion des secrets

Session de l'après-midi :

- Flux d'authentification : login, rotation de token, logout et stockage sécurisé côté client
- Contrôles d'accès : middleware d'autorisation, scopes/rôles et sécurisation des ressources
- Gestion des fichiers (option) : upload, limites, type checking et stockage (local/cloud)

TP / Exercice :

- Implémenter l'authentification et l'autorisation sur le fil rouge : endpoints auth, JWT + refresh, middleware RBAC, validation d'entrées et rate limiting. Ajouter des tests d'intégration sur les parcours sécurisés. Livrable : API sécurisée (auth + RBAC) avec collection de tests et preuves de conformité.

Points clés & takeaways :

- Concevoir un modèle d'authentification/autorisation robuste et auditable
- Appliquer les protections essentielles d'une API Express en contexte entreprise
- Sécuriser des parcours critiques et valider via tests d'intégration

Jour 4 – React professionnel : état, données, performance et expérience utilisateur

Session du matin :

- Architecture front : features, composants partagés, design system (notion) et conventions

- Gestion d'état et données : server state vs UI state, cache, invalidation, gestion erreurs/chargements
- Formulaires pro : validation, accessibilité, états transitoires et patterns réutilisables

Session de l'après-midi :

- Performance : profiling, mémoïsation, virtualisation de listes et optimisation du rendu
- Routing avancé : guards côté UI, routes protégées, lazy loading et gestion des permissions
- Qualité UX : messages, empty states, i18n (notion) et accessibilité opérationnelle

TP / Exercice :

- Intégrer le front fil rouge avec l'API sécurisée : routes protégées, gestion de session, cache des requêtes et gestion des erreurs. Optimiser un écran lourd (profiling + correctifs) et rendre un formulaire accessible (labels, erreurs, focus). Livrable : front opérationnel, performant et cohérent avec le contrat d'API.

Points clés & takeaways :

- Structurer un front React scalable et aligné sur des features
- Optimiser les performances et fiabiliser les états (loading/erreur/cache)
- Mettre en place des parcours UX accessibles et des routes protégées

Jour 5 – Tests, CI/CD, déploiement et observabilité**Session du matin :**

- Stratégie de tests : pyramide, unitaires, intégration, E2E; données de test et isolation
- Tests API : supertest, contrats, mocks et tests de sécurité basiques
- Tests front : Testing Library, scénarios critiques et anti-patterns

Session de l'après-midi :

- CI/CD : pipelines, quality gates, build multi-env, gestion des secrets et artefacts
- Déploiement : Docker (recommandé), variables d'environnement, reverse proxy et migration de données (notion)

- Observabilité : logs structurés, tracing/métriques simples, monitoring erreurs et runbook

TP / Exercice :

- Finaliser le fil rouge : ajouter suites de tests (API + front) et un pipeline CI minimal (lint, tests, build). Conteneuriser (ou packager) l'application, préparer un déploiement cible et instrumenter logs/erreurs. Livrable : application prête à livrer (tests + pipeline + artefacts) et dossier d'exploitation (runbook).

Points clés & takeaways :

- Mettre en œuvre une stratégie de tests complète et mesurable
- Concevoir un pipeline de livraison robuste et reproductible
- Préparer l'exploitation avec observabilité et documentation opérationnelle

Organisation

Formateur

Les formateurs de Docaposte Institute sont des experts de leur domaine, disposant d'une expérience terrain qu'ils enrichissent continuellement. Leurs connaissances techniques et pédagogiques sont rigoureusement validées en amont par nos référents internes. Riches de leur expérience sur le sujet, ils sauront accompagner vos collaborateurs dans leur montée en compétences.

Moyens pédagogiques et techniques

- Apports des connaissances communes.
- Mises en situation sur le thème de la formation et des cas concrets.
- Méthodologie d'apprentissage attractive, interactive et participative.
- Equilibre théorie / pratique : 60 % / 40 %.
- Supports de cours fournis au format papier et/ou numérique.
- Ressources documentaires en ligne et références mises à disposition par le formateur.
- Pour les formations en présentiel dans les locaux mis à disposition, les apprenants sont accueillis dans une salle de cours équipée d'un réseau Wi-Fi, d'un tableau blanc ou paperboard. Un ordinateur avec les logiciels appropriés est mis à disposition (le cas échéant).

Dispositif de suivi de l'exécution et de l'évaluation des résultats de la formation

En amont de la formation :

- Recueil des besoins des apprenants afin de disposer des informations essentielles au bon déroulé de la formation (profil, niveau, attentes particulières...).
- Auto-positionnement des apprenants afin de mesurer le niveau de départ.

Tout au long de la formation :

- Évaluation continue des acquis avec des questions orales, des exercices, des QCM, des cas pratiques ou mises en situation...

A la fin de la formation :

- Auto-positionnement des apprenants afin de mesurer l'acquisition des compétences.
- Evaluation par le formateur des compétences acquises par les apprenants.
- Questionnaire de satisfaction à chaud afin de recueillir la satisfaction des apprenants à l'issue de la formation.
- Questionnaire de satisfaction à froid afin d'évaluer les apports ancrés de la formation et leurs mises en application au quotidien.

Accessibilité

Nos formations peuvent être adaptées à certaines conditions de handicap. Nous contacter pour toute information et demande spécifique.