

PRÉNOM NOM

DÉVELOPPEUR FULL-STACK REACT / NODE.JS, PLATEFORMES SAAS ET API À FORT TRAFIC

9 ans d'expérience full-stack, front et back	1,2 M requêtes API absorbées par jour	÷ 5 sur le temps de réponse médian de l'API
84 % de couverture de tests sur le back-end	850 000 visiteurs par mois sur le tunnel d'achat refondu	2 h de la merge request à la production, contre 5 jours

PROFIL

Je construis des produits web depuis 9 ans, du premier écran React jusqu'aux API qui tiennent la charge un lundi matin. 3 produits livrés en production : une plateforme SaaS logistique qui absorbe 1,2 million de requêtes par jour, un tunnel d'achat e-commerce traversé par 850 000 visiteurs par mois, un portail assuré ouvert à 40 000 clients. Mon terrain : React et TypeScript côté front, Node.js ou Java côté back, PostgreSQL, le tout déployé sur AWS avec du Terraform et un pipeline qui ne demande la main à personne. Je travaille dans des équipes produit de 5 à 10 personnes, avec un PO et un designer, pas en retrait derrière un ticket. Les tests, je les écris pendant, pas après : mes projets tournent entre 70 et 85 % de couverture. Ce qui m'intéresse dans une mission, c'est de comprendre ce que le produit doit changer pour le métier avant d'ouvrir l'éditeur.

COMPÉTENCES CLÉS

React 18, hooks, React Query, Next.js · TypeScript, typage strict et refactoring à grande échelle · Node.js, Express, NestJS · Java 17, Spring Boot · Conception d'API REST et GraphQL, documentation OpenAPI · PostgreSQL, modélisation, indexation, optimisation de requêtes	Redis, cache applicatif et files de traitement · Kafka, traitement d'événements asynchrones · Docker, Docker Compose, Kubernetes · AWS (ECS, S3, RDS, Lambda, CloudWatch) · Terraform, infrastructure décrite en code · CI/CD avec GitLab CI et GitHub Actions	Tests unitaires Jest et Vitest, tests d'intégration · Tests end to end Cypress et Playwright · Architecture hexagonale et découpage en domaines · Observabilité : Datadog, Grafana, Sentry · Revue de code, binômage et accompagnement de développeurs juniors · Scrum et Kanban : refinement, estimation, démonstration client
--	--	---

EXPÉRIENCE PROFESSIONNELLE

Développeur full-stack senior, plateforme SaaS de pilotage logistique janvier 2023 à aujourd'hui

[Nom du client] édite une plateforme SaaS de gestion de flotte et de tournées, utilisée par 340 entreprises de transport en France et en Espagne. Je suis intégré à une équipe produit de 9 personnes : 6 développeurs, 1 PO, 1 designer, 1 testeur, en Scrum avec des sprints de 2 semaines. L'enjeu tenait en une phrase : absorber un trafic passé de 200 000 à 1,2 million de requêtes API par jour en 18 mois sans dégrader les temps de réponse contractualisés. J'interviens sur toute la chaîne, du composant React jusqu'aux traitements nocturnes qui recalculent les tournées.

- Concevoir et développer le module de planification des tournées en React 18 et TypeScript, utilisé chaque jour par 4 500 exploitants pour affecter 60 000 livraisons.
- Refondre l'API de calcul d'itinéraires en Node.js et NestJS afin de ramener le temps de réponse médian de 900 ms à 180 ms sous charge réelle.
- Mettre en place la stratégie de tests de l'équipe, soit 1 400 tests unitaires Jest et 90 scénarios Playwright rejoués à chaque merge request.
- Industrialiser le déploiement sur AWS ECS avec Terraform et GitLab CI pour faire passer le cycle de mise en production de 5 jours à 2 heures.
- Encadrer 2 développeurs juniors au quotidien : revue de code systématique, binômage 3 demi-journées par semaine, montée en compétence sur TypeScript et les tests.

Résultats : Temps de réponse médian de l'API divisé par 5, de 900 ms à 180 ms, ce qui a permis de signer 3 comptes grands transporteurs dont l'appel d'offres exigeait un SLA à 300 ms. · Incidents de production ramenés de 24 à 6 par trimestre après la mise en place des tests end to end et de l'alerting Datadog.

Environnement technique : React 18 · TypeScript · Node.js · NestJS · PostgreSQL 15 · Redis · AWS ECS · Terraform · Playwright

Développeur full-stack, refonte du tunnel d'achat e-commerce septembre 2020 à décembre 2022

[Nom du client] vend en ligne de l'équipement sportif : 850 000 visiteurs uniques par mois, 45 millions d'euros de chiffre d'affaires annuel, 3 pays. J'ai rejoint la squad Checkout, 5 développeurs et 1 PO, pour reprendre un tunnel d'achat vieux de 7 ans dont le taux d'abandon dépassait 72 % sur mobile. L'équipe fonctionnait en Kanban, avec plusieurs mises en production par jour et aucune tolérance sur les régressions de paiement. Le back-end existant tournait en Java, le front était à reconstruire depuis une page blanche.

- Reconstruire le tunnel d'achat en React et Next.js, du panier à la confirmation, livré par étapes sur 10 % puis 50 % puis 100 % du trafic.
- Développer les services de commande et de paiement en Java 17 et Spring Boot, avec l'intégration de 3 prestataires de paiement et la gestion des reprises sur échec.
- Instrumenter 22 événements analytiques dans le parcours pour mesurer la friction étape par étape et arbitrer les priorités avec le PO.
- Optimiser les performances du front : score Lighthouse mobile passé de 38 à 91, affichage du panier ramené sous 1,5 seconde en 4G.
- Piloter les tests A/B côté technique, 14 expérimentations menées sur 18 mois, dont 5 généralisées.

Résultats : Taux de conversion du tunnel passé de 1,8 % à 2,0 %, soit 5 millions d'euros de chiffre d'affaires additionnel sur 12 mois à volume de trafic constant. · Abandons à l'étape de paiement réduits de 31 % après la refonte de la saisie de carte et l'ajout du paiement en 3 fois.

Environnement technique : React · Next.js · TypeScript · Java 17 · Spring Boot · PostgreSQL · Kafka · Docker · GitHub Actions

Développeur back-end puis full-stack, API partenaires et portail assuré juin 2017 à août 2020

[Nom du client] est un assureur spécialisé qui gère 210 000 contrats et travaille avec un réseau de courtiers. J'arrive dans une équipe de 4 développeurs avec une mission claire : ouvrir un système de gestion Java vieux de 12 ans à des partenaires externes, sans le réécrire. Sprints de 3 semaines, recette encore manuelle, 2 mises en production par trimestre au démarrage. J'ai tenu le back-end la première année, puis pris en charge le front du portail assuré à partir de la deuxième.

- Développer 28 endpoints REST d'accès aux contrats et aux sinistres, documentés en OpenAPI et consommés par 9 cabinets de courtage.
- Extraire 3 domaines métier du monolithe vers des services Node.js autonomes, en bascule progressive et sans interruption de service.
- Construire le portail assuré en React, adopté par 40 000 clients la première année, avec un parcours de déclaration de sinistre en 5 écrans.
- Automatiser la recette avec 320 tests d'intégration exécutés chaque nuit, en remplacement de 2 jours de tests manuels par sprint.
- Former 4 développeurs de l'équipe aux tests automatisés et à la revue de code lors d'ateliers mensuels.

Résultats : Fréquence de mise en production passée de 2 par trimestre à 1 par semaine grâce à l'automatisation de la recette. · Appels au service client réduits de 18 % la première année après l'ouverture du portail en libre service.

Environnement technique : Java 11 · Spring Boot · Node.js · React · Oracle · REST et OpenAPI · Jenkins · Docker

FORMATION

Diplôme d'ingénieur, spécialité informatique et systèmes d'information · [Nom de l'école] 2014 à 2017

Licence informatique · [Nom de l'université] 2011 à 2014

CERTIFICATIONS

- AWS Certified Developer, Associate
- Certified Kubernetes Application Developer (CKAD)
- Professional Scrum Master I (PSM I)

LANGUES

Français, langue maternelle · Anglais, courant professionnel (C1), réunions quotidiennes et documentation technique en anglais depuis 4 ans