



SÉBASTIEN NOBOUR

Tel: +33780988325
E-mail: sebastien@devedanos.com
LinkedIn: [sebastien-nobour](#)

Site web: devedanos.com

DÉVELOPPEUR FULL-STACK FREELANCE NODEJS NEXTJS AWS

PROFIL

J'aide les entreprises à créer des applications sur mesure. Cela fait 3 ans que j'interviens sur le développement d'Interfaces Utilisateur, design et développement d'APIs backend et base de donnée, déploiement chez hébergeurs web.

Atouts: anglais et français courant, mindset action-driven, data-driven et client-first;

EXPÉRIENCE

DÉVELOPPEUR FULL-STACK

Freelance (depuis 2020)

- Product feedback: développement d'une SPA en ReactJS et API GraphQL en NestJS
- ISCO (Pays-Bas): mise à jour d'une SPA en EmberJS et API REST en Symfony d'un CRM, développements des tests

DÉVELOPPEUR PHP

Valeo (2019 - 2020)

- Maintenance d'une application de monitoring de licence
- Veille technologique, développement de Proof Of Concepts, propositions techniques

DÉVELOPPEUR WEB JUNIOR

Freelance (2019)

- Refactory Devops (Pays-Bas): Maintenance et développement de sites et applications web

FORMATIONS

- Licence Professionnelle Développeur & Concepteur d'Applications chez My Digital School (2019 - 2020)
- Certifications développeur et intégrateur web Web Force 3 (Sept. 2018 - Janv. 2019)
- BTS Domotique IFA Delorozoy (2012- 2014)

CENTRES D'INTÉRÊT

- Développement personnel, leadership, communication, techniques de lecture rapide;
- Entrepreneuriat, crypto-monnaies, e-commerce via Amazon FBA;
- Mode de vie digital nomad;
- Sports: gymnastique artistique, musculation, football, basketball, natation, vélo;
- Depuis 2018 : bénévole dans une association anglophone;
- Bénévole à temps plein de 2015 à 2017: enseignant individuel, orateur;

COMPÉTENCES

- Architecture et concepts: MVC, FLUX, Data Down Actions UP (DDAU), POO, SOLID patterns, REST and JSON APIs and GraphQL, SPA (Server Side Rendering and Client Side Rendering), JAMStack for static website, websocket, web worker, micro-services, serverless, containerization, virtualization, Infrastructure As Code
- Systèmes : Linux (Ubuntu, Debian, Raspbian), Mac OS, Windows
- Langages: JavaScript, TypeScript, PHP7, SQL, HTML5, CSS3, XML, SASS, PostCSS
- Frameworks CSS: Bootstrap [4,5], TailwindCSS 2, Bulma CSS, Material UI, AntDesign
- Frameworks & librairies JS Frontend: ReactJS(Redux, React Apollo, React Router), EmberJS, JQuery, Webpack
- Frameworks JS Backend: NodeJS, ExpressJS, NestJS(Apollo-graphql), SocketIO
- Frameworks JS Logiciels: ElectronJS
- Frameworks PHP: Symfony (4.5), Twig, API-Platform
- Bases de données et outils : PostgreSQL, MySQL, OracleSQL, MongoDB, DoctrineORM, TypeORM, Mikro-ORM, Mongoose
- Outils de testing et qualité: TDD, ESLint, Prettier, Jest, Mocha, Chai HTTP, Qunit, Coverall, PHPUnit, Cypress, BrowserStack
- Versionning: Git, Github, Gitlab, Bitbucket
- CI/CD: Gitlab.ci, Github Actions, Travis CI, AWS Code Pipeline
- Devops: GitlabCI, Github Actions, Docker, Serverless Framework
- Cloud: Amazon Web Services (Lambda, ECS, ECR, EC2, S3, EFS, Amplify, Route53, LightSail, IAM, SNS, SES, SQS, API Gateway, DynamoDB, CloudWatch, Secret Manager, IAM, S3, DynamoDB, BitBucket, CloudFormation, API Gateway, EventBridge, Code Pipeline)
- Gestion de projet et outils: Github Projects, Gitlab, CollabNet, Jira, Confluence
- Outil de design visuel: Figma, Adobe XD
- CMS: Shopify, Wordpress/Bedrock, WooCommerce

SIDE PROJECTS

- nobour-eshop.com: application e-commerce en MERN Stack, AWS (EC2, Amplify), Docker, Github Actions;
- crm.devedanos.com application CRM: PHP, API-Platform, React.js;
- sym-5-bnb.devedanos.com: application réplique d'Air B n'B: PHP, Symfony;
- applications todo list et school manager: NestJS, ReactJS, MongoDB, PostgreSQL, APIs GraphQL et REST, AWS, TypeScript;
- application d'enchères: Node.js, React.js, Serverless Framework AWS, AWS Lambda;

PRODUCT FEEDBACK

MARS 2021

DÉVELOPPEUR FULL-STACK FREELANCE NODEJS REACTJS NESTJS TYPESCRIPT

Introduction:

Product Feedback est une startup créée en 2019 à Marseille (13, France) qui ambitionne de révolutionner la gestion des retours clients sur un produit. Elle s'inspire de l'application notion.so.

Product Feedback est parti du constat que la collecte et la synthèse des retours utilisateurs sur un produit était difficile et chronophage : informations non partagées avec l'ensemble de l'équipe, fiches ou documents justificatifs manquants.... Pour simplifier cette gestion, Product Feedback développe une interface collaborative en ligne, rassemblant en un seul endroit les réponses des clients aux formulaires, les rapports des membres de l'équipe produit sur ces réponses, et les décisions finales basées sur ces rapports. Elle prévoit d'ajouter des services supplémentaires tels que l'intégration de feuilles de calcul Excel et plus encore...

En mars, son équipe était composée d'un CEO, d'un CMO basé à Marseille et d'un CTO travaillant depuis la République tchèque. Je travaillais directement avec le CTO.

Contexte:

Product Feedback m'a proposé de travailler pour eux en tant que freelance au lieu de passer des tests techniques, avant de m'embaucher. C'était un poste en full-remote.

Lors du 6ème jour d'essai, j'ai eu mon premier entretien avec le CEO. Durant ce dernier, j'ai appris que l'équipe de Product Feedback recherchait en fait un développeur senior pour substituer son CTO. J'ai donc travaillé 6 jours pour Product Feedback.

Tâches:

- Participation dans l'implémentation de l'architecture de l'application, à la conception de la base de données et des entités backend.
- Développement d'une API GraphQL ainsi que de ses requêtes et mutations
- Développement d'une interface utilisateur sous forme Single Page Application, avec ses formulaires et ses composants réactifs
- Développement d'une API GraphQL, avec ses resolvers, ses services, ses repositories.

Technologies:

NodeJS, TypeScript, NestJS, ReactJS, React-Apollo, PostgreSQL, AWS (Amplify), git, Github, Docker, styled-components, AntDesign, Mikro-ORM, Material UI, Responsive Web Design, styled-components

DÉVELOPPEUR FULL-STACK FREELANCE SYMFONY EMBERJS

Introduction:

ISCO est une entreprise néerlandaise qui fabrique des chaussures de sécurité. Elle dispose d'un CRM développé par Refactory Devops, une société unipersonnelle néerlandaise.

Contexte:

Refactory Devops a fait appel à mes services pour mettre à jour le CRM d'ISCO.

J'ai travaillé sur la mise à jour de son backend et de son frontend. Refactory Devops s'occupe des déploiements dans les environnements de staging et de production, et de la revue de code. Le suivi des tickets se fait sur Gitlab.

Tâches:

- Mise à jour de l'API REST, de Symfony 3 à 4 et adaptation de la configuration de ses conteneurs de services
- Mise à jour de la Single Page Application d'EmberJS 2 à 3, avec sa syntaxe Octane dans les templates, le remplacement des classes ember par des classes JS natives dans les composants, controllers, helpers, models.
- Développement de tests unitaires et d'intégration descriptifs avec PHPUnit pour l'API REST, et Qunit pour la SPA.

Technologies:

PHP 7, Javascript, EmberJS (2, 3), Qunit (qunit-dom), PHPUnit, CI/CD, Gitlab CI, Docker, NodeJS, YAML, Symfony (3, 4);

DÉVELOPPEUR PHP

Introduction:

Valeo est une grande entreprise internationale, fondée en France, et l'une des leaders du marché de l'équipement automobile. Elle est composée de 5 activités : systèmes thermiques, systèmes de visibilité, systèmes de propulsion, systèmes de confort et d'aide à la conduite, et activités de deuxième monte. Chaque activité dispose d'une équipe Méthodes et Outils, qui dispose elle-même d'une équipe Engineering Information System (EIS). L'équipe EIS gère, entre autres, les licences des logiciels, permettant ainsi aux ingénieurs de disposer des outils numériques nécessaires à leur travail (conception, modélisation 3D, bancs d'essais virtuels).

Contexte:

Durant mon apprentissage, j'ai rejoint l'équipe EIS du groupe systèmes thermiques. Directement sous la supervision de mon responsable et en tant que seul développeur sur le projet, j'étais en charge de la maintenance de Licence Server Status, une application PHP personnalisée permettant de surveiller les 400 serveurs de Valeo dans le monde entier afin de vérifier le statut des licences logicielles. Cette application stocke les informations dans une base de données, puis les affiche sur un tableau de bord, fournissant des alertes et envoyant des emails lorsqu'une licence est sur le point d'expirer. Ce projet a connu plusieurs versions en 15 ans, et a souvent été réécrit de zéro par les apprentis successifs. Le but de ce projet est d'aider EIS dans la prévention du renouvellement et de l'achat de licences logicielles afin de mieux gérer les budgets de Valeo.

Tâches:

- Suggestion et mise en œuvre de pratiques de développement, d'architecture et de technologies afin de préserver le projet et de ne plus avoir à le réécrire (utilisation du versioning, déploiement des programmes via git au lieu de ftp, utilisation d'un système de gestion des tickets, passage du projet à OOP et MVC, ajout de tests automatisés et CI/CD...)
- Développement et maintenance de l'interface utilisateur
- Développement de graphiques avec ChartJS
- Définition, développement et maintenance de scripts PHP interrogeant les serveurs et leurs daemons de licences logicielles
- Conception de l'architecture et de l'implémentation de la base de données
- Développement des requêtes SQL
- Veille technologique sur Symfony 5 en développant une réplique d'Air B n'B accessible sur sym-5-bnb.devedanos.com
- Déploiement dans des environnements de pré-production et de production
- Développement de Proof of Concepts pour l'application PHP personnalisée License Server Status sur Symfony 5 (non réalisé en raison de la pandémie de COVID-19)

Technologies:

PHP (5, 7), Symfony (4, 5), BeyondCompare, Git, CollabNet, Javascript, AJAX, HTML, CSS, Bootstrap, Apache, CentOS, Windows Server 2008 et 2012, Windows 10, Scheduled task, WinBash, YAML, Bootstrap, Bootswatch, Doctrine ORM, SQL, OracleSQL, MySQL, OVH

DÉVELOPPEUR WEB JUNIOR FREELANCE

Introduction:

Refactory Devops est une entreprise unipersonnelle néerlandaise de services numériques. Son développeur a plus de 15 ans d'expérience. Il développe ses propres solutions devops, construit des sites web et des applications pour ses clients.

Contexte:

En 2019, j'avais l'intention de déménager aux Pays-Bas. Pendant quelques mois, j'ai travaillé aux côtés du développeur de Refactory Devops sur des projets pour ses clients, ainsi que sur son propre projet de startup. Je travaillais à la fois depuis la France et depuis Nimègue aux Pays-Bas.

Tâches:

- Implémentation de l'internationalisation de la langue d'une Single Page Application
- Développement du système d'authentification via JWT et OAuth2
- Intégration web pour les sites vitrines via CMS
- Maintenance et correction des bugs sur Single Page Applications
- Migration de sites web d'un service d'hébergement web à un autre

Technologies:

Neos Framework, Neos CMS, EmberJS 2, l18n, Javascript, HTML5, CSS3, PHP 7, YAML, cPanel;

DEVEDANOS.COM

JUILLET 2021

Développement frontend avec EmberJS

Introduction:

Ce projet correspond au site web portfolio de ma micro entreprise pour son activité liée au développement web. Ce site est disponible sur devedanos.com.

Il s'agit d'une Single Page Application basée sur EmberJS, utilisant la technique du Server Side Rendering grâce à l'extension ember-fastboot. Mais elle est également capable d'utiliser la génération de sites statiques via JAMStack grâce à l'extension empress. J'utilise le Server Side Rendering pour apporter une meilleure UX et un bon SEO à mon site. Il est déployé sur Netlify. Le but de ce site est de rendre visible mon activité de développement web enregistrée sous la marque "Devedanos".

Tâches:

- Organisation de mes tâches sur notion.so
- Préparation du design de l'UI : choix des couleurs, logo, dessin des composants
- Dépôt de la marque à l'INPI
- Développement des composants de l'interface utilisateur et internationalisation
- Développement de fonctionnalités par TDD
- Déploiement en production
- Référencement (utilisation de mots-clés, HTML sémantique, scripts différés)

Technologies:

EmberJS (fastboot, empress, intl), Netlify, Handlebars, Javascript, NodeJS, Tailwind CSS, PostCSS (syntax SASS extension), NotionsSO, git, github, Responsive Web Design, eslint, Plausible, Google (Search Console and Trends), Netlify

Développement gull-stack MERN stack et Docker

Introduction:

Il s'agit de mon projet personnel le plus abouti, déployé sur nobour-eshop.com, réalisé en près de 400 heures de travail.

Afin d'améliorer mes compétences sur la MERN stack, j'ai suivi la formation "Mern eCommerce from scratch" sur Udemy. Ayant estimé que cette formation n'allait pas aussi loin que je le souhaitais, ni assez vite, ni que l'application produite était utilisable en conditions réelles, j'ai ajouté des fonctionnalités et modifié la structure du projet par moi-même.

Cette application permet à un commerçant de mettre plusieurs produits en vente et d'être payé via Paypal. Un système d'emailing envoie des emails au propriétaire de la boutique lorsque le stock de produits est faible, lorsqu'une commande doit être préparée. Des emails sont également envoyés aux clients pour confirmer la création du compte, la modification du mot de passe, la confirmation de l'achat, l'envoi de sa commande. Une facture est automatiquement générée à l'achat et est téléchargeable à la fois par le commerçant et l'acheteur. Seule l'expédition n'est pas gérée par l'application.

Tâches:

Du côté du backend, cette formation était une révision. Par contre, sur le frontend, cette formation m'a permis de :

- prendre en main Redux pour gérer l'état global (store) du SPA en ReactJS
- mettre en place des actions et des reducers afin d'exécuter des requêtes HTTP CRUDS vers l'API REST, récupérer le panier de produits depuis le stockage local
- mettre en place une restriction d'accès aux vues réservées à l'administrateur et des redirections utilisant les hooks de React Router

Pour ma part, voici ce que j'ai ajouté au projet, ou même appris à l'occasion :

- Développement de la génération de factures en PDF
- Développement de la gestion des stocks de produits, annulation des commandes
- Développement d'emails aux clients pour confirmer la création de leur compte, réinitialiser leur mot de passe, les informer de la confirmation d'achat et de l'expédition de leurs commandes
- Envoi d'emails au marchand pour l'informer des nouvelles commandes, et que le stock de produits est faible
- Création d'images Docker, mise en service et déploiement de conteneurs Docker dans AWS ECR, EC2
- Mise en place d'un pipeline CI/CD à plusieurs étapes et d'environnements de production et de staging via des actions Github pour le backend et un pipeline CI/CD via AWS Amplify pour le frontend.
- Sécurisation de l'API REST ExpressJS avec le middleware Express CORS et helmet
- Création d'une infrastructure pour l'écosystème de l'application sur AWS
- Création de bases de données pour tous les environnements sur les services MongoDB Atlas et conteneurs Docker

Technologies:

MongoDB (Compass, Atlas), ExpressJS, ReactJS, NodeJS, Docker (docker-compose), Github Actions, AWS (Amplify, ECS, EC2, ECR, Route53, VPC, IAM, EFS), AWS SDK, helmet, ssl, multer, morgan, mongoose, axios, Responsive Web Design, Redux, POSTMAN, Bootstrap (bootswatch themes);

Développement backend cloud Serverless AWS Lambda NodeJS

Introduction:

Afin d'améliorer mes compétences sur AWS et de m'assurer que j'avais bien compris son utilisation après avoir suivi le cours "AWS - Les 12 services fondamentaux du Cloud n°1", j'ai suivi la formation "Serverless Framework bootcamp : NodeJS, AWS & Microservices", toutes deux sur Udemy. Cette dernière m'a permis de développer une application de vente aux enchères, en provisionnant des ressources sur AWS grâce à Serverless Framework. J'ai développé 3 micro-services : les services d'authentification, d'enchères et de notification et les ai connecté à une Single Page Application en React.js.

Le code source est réparti sur 3 repositories github:

- <https://github.com/daproclima/aws-sls-notification-service>
- <https://github.com/daproclima/aws-sls-auth-service>
- <https://github.com/daproclima/aws-sls-auction-service>

Tâches:

- Développement des endpoints HTTP effectuant des CRUD via API Gateway, appelant à leur tour des Lambda functions
- Utilisation de DynamoDB comme SGBD afin d'effectuer un CRUD sur les données et créer des schémas de documents pour la validation des données et la gestion des erreurs
- Mise en place de files d'attente des messages pour la communication entre micro-services via AWS Simple Queue Service
- Mise en place d'événements planifiés via AWS EventBridge afin de clore une enchère au bout d'une heure.
- Gestion de toute l'architecture des micro-services via AWS CloudFormation indirectement exploité par les fichiers serverless.yaml de Serverless Framework (interconnexion des services par la récupération de leurs arn, url, environnement...)
- Mise en place du stockage d'objets via AWS S3
- Mise en place d'envoi de notifications par emails via AWS SES permettant d'informer si l'enchère a été remportée par quelqu'un ou n'a pas trouvé preneur
- Mise en place d'un middleware d'authentification et d'autorisation en JWT via API Gateway Lambda Authorizer, optimisant les coûts si le même JWT a été utilisé il y a moins de 300 secondes sur le même endpoint API Gateway

Technologies:

NodeJS, ReactJS, JWT, HTTP, Serverless Framework, YAML, AWS (Lambda functions, IAM, S3, DynamoDB, BitBucket, CloudFormation, API Gateway, EventBridge, Simple Notification Service, Simple Queue Service, Identity and Access Management), git, Github

Développement backend NodeJS NestJS TypeScript

Introduction:

Afin d'en savoir plus sur NestJS et de m'assurer que mon passage de PHP et Symfony à NodeJS était un bon choix, j'ai suivi une courte formation en ligne sur Udemy : "NestJS Zero to Hero - Modern TypeScript Back-end development".

Elle comprenait deux projets pour créer une liste de choses à faire et un système de gestion des classes d'école. J'ai développé une API REST utilisant PostgreSQL pour la todo-list et une API GraphQL, ses mutations et ses requêtes utilisant MongoDB comme SGBD pour le système de gestion des classes

Tâches:

- Développement de tests unitaires avec Jest
- Conception et développement d'API REST et GraphQL effectuant des opérations CRUD
- Authentification et autorisation pour les applications backend,
- Développement de scripts avec TypeORM pour interagir avec les bases de données MongoDB et PostgreSQL
- Hachage de mots de passe et stockage d'informations sensibles
- Déploiement des API en production sur Amazon Web Services.
- Mise en œuvre d'un logging efficace dans les applications backend
- Mise en œuvre de la validation des données et utilisation de Pipes
- Modélisation des entités pour la couche de persistance
- Développement d'objets de transfert de données (DTO)

Technologies:

NodeJS, TypeScript, NestJS, ReactJS, Apollo-server, PostgreSQL, git, Github, TypeORM, HTTP, JWT, Jest, SSL, MongoDB, Postman, PgAdmin, Jest;

Développement full-stack ReactJS API-Platform

Introduction:

Afin de savoir comment développer avec React et créer une API REST avec PHP, j'ai suivi la formation "Modern development with ApiPlatform (Symfony) and React" de Lior Chamla pendant un mois. Grâce à cette formation, j'ai développé une application de gestion de la relation client, composée d'une Single Page Application en ReactJS et d'une API REST basée sur API-Platform, respectant la spécification API JSON.

Elle est déployée sur Heroku et accessible sur crm.devedanos.com.

Tâches:

- Développement du SPA et des formulaires
- Développement et configuration de l'API JSON
- Mise en œuvre de l'architecture de la base de données
- Développement de scripts et de requêtes pour le filtrage, l'ordonnancement, la pagination
- Gestion de la sérialisation/désérialisation des données
- Développement de la sécurité : validation des données, authentification par JWT
- Gestion manuelle de la mise en cache des données côté client
- Déploiement de l'ensemble de l'application sur Heroku

Technologies:

PHP 7, Symfony 5, API-Platform, Axios, React (React-router, react toastify, react bootstrap...), HTTP, Bootstrap (bootswatch theme), HTML5, CSS3, DoctrineORM, SQL, MySQL, Heroku;

Développement full-stack Javascript NodeJS

Introduction:

Dans le cadre de ma formation au titre de Concepteur Développeur d'Applications à My Digital School Paris, j'étais dans une équipe de 5 personnes composée d'un web designer, de 2 web marketeurs, et d'un autre développeur web. Notre projet a permis aux webdesigners et marketeurs de notre équipe de valider leur année scolaire.

J'ai été nommé chef de projet par l'équipe et j'avais également le rôle de lead developer. Cela m'a permis de développer des compétences en marketing, web design, gestion de projet.

L'objectif de notre projet " Worker Genius " était de faciliter la reconnaissance des compétences des professionnels du numérique et leur recherche d'emploi. Pour ce faire, nous avons commencé le développement d'une application étant à la fois une plateforme de test technique, un job board et un système de chat en temps réel entre recruteurs et professionnels du numérique. Concernant le développement, je me suis principalement concentré sur le backend, laissant la majorité du développement du frontend à l'autre développeur de l'équipe.

L'application était composée de plusieurs pages HTML rendues par un serveur ExpressJS et d'une API REST différente.

Tâches:

- Définition des choix technologiques, structure du projet
- Définition et développement de la structure du modèle de base de données
- Développement d'une API REST connectée à mongoDB, validant également les payloads JSON
- Développement de l'authentification avec JWT
- Développement d'un serveur websocket pour un service de chat
- Développement d'une partie de l'interface utilisateur
- Déploiement sur PlanetHoster

Technologies:

AGILE, Trello, NodeJS, ExpressJS (express-validation), HTML5, CSS3, MongoDB, HTTP, HandlebarsJS, Mongoose, SocketIO, Bootstrap 4, axios, Adobe XD