



## FREELANCE AGILE NODEJS JAVASCRIPT AWS DEVELOPER

### PROFILE

I help companies create custom applications and prioritize the development of high value-added features with Javascript technologies (Node.js, React.js, Next.js, Nest.js...).

I have been working for 4 years on the development of User Interfaces, design and development of backend and database APIs, deployment at web hosting companies.

Assets: fluent English and French (native), action-driven, data-driven and client-first mindset, polyvalent, proactive, constructive communication, solid experience on Javascript

### EXPERIENCE

#### FULL-STACK DEVELOPER

##### Freelance (since 2020)

- Weekandgo.com: development of a niche marketplace web application using Next.js, API Platform, Strapi CMS linked to a blog and business API for Rapido Group
- Product feedback: development of a SPA using ReactJS and a GraphQL API in Nest.js
- Refactory Devops (Netherlands): maintenance of a CRM by upgrading a SPA using EmberJS and a REST API in Symfony and development of tests

#### PHP DEVELOPER

##### Valeo (2019 - 2020)

- Maintenance of a license monitoring application, technology watch, development of Proof Of Concepts, technical proposals

#### JUNIOR WEB DEVELOPER

##### Freelance (2019)

- Refactory Devops (Netherlands): Maintenance and developpement of websites and web applications

#### PREVIOUS JOBS

- IT support technician at Spie in 2018
- Technical support and logistics technician at Logicom in July 2017
- Fast food team member at Burger King in 2016
- Itinerant technician on electronic security systems for Automatic Alarm in 2015
- Itinerant technician on automated opening systems at Automatismes et Portes pour le Bâtiment from 2012 to 2014

### EDUCATION

- Training and coaching for freelance web developers with Mike Codeur and Charlo Allizard (2021)
- Entrepreneurship and marketing training and coaching by Alec HENRY with Entrepreneurs.com program (2020)
- Training and coaching in e-commerce, entrepreneurship and affiliate marketing by Olivier ALLAIN with the Amazon Revolution and Marketing Revolution programs (2020)
- Bachelor (EQF\* Level 6) Application Developer at My Digital School (2019 - 2020)
- Web developer and integrator certifications at Web Force 3 (Sept. 2018 - Janv. 2019)
- BTS Domotique (EQF\* Level 5) at IFA Delorozoy (2012- 2014)

- Architecture and 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, ORM/ODM, TDD, BDD, testing (unit, integration, interactive, visual, application, system)
- OS: Linux (Ubuntu, Debian, RaspBian), Mac OS, Windows
- Languages: JavaScript ES5, PHP7-8, SQL, HTML5, CSS3, XML, SASS, PostCSS
- CSS frameworks: Bootstrap [4,5], TailwindCSS 2, Bulma CSS, Material UI/MUI, AntDesign
- JS librairies & frameworks: React.js (16, 17 and Redux, React Apollo, React Router, Jotai, React Context), Ember.js, JQuery, Next.js (12, 13), Lodash, React Hook Form, Formik, Joi, emotion, luxon, svggo;
- Module bundlers: Webpack, SWC, Babel;
- Node.js backend frameworks: Node.js, Express.js, Nest.js (Apollo-graphql), SocketIO, Koa.js
- Node.js software frameworks: Electron.js
- PHP frameworks: Symfony (4,5), API-Platform
- Databases and tooling: PostgreSQL, MySQL, OracleSQL, MongoDB, DoctrineORM, TypeORM, Mikro-ORM, Mongoose
- Quality, tests and tooling: ESLint, Prettier, Stylelint, Jest, Mocha, Chai HTTP, Qunit, Coverall, PHPUnit, Cypress, BrowserStack, Nightwatch, Playwright, testing-library, Cucumber, Gherkin, Husky, MirageJS
- Versionning: Git, Github, Gitlab, Bitbucket
- CI/CD: Gitlab CI, Github Actions, Travis CI, AWS Code Pipeline
- Devops: Docker and Docker compose, Serverless Framework
- Cloud: Amazon Web Services (Lambda, ECS, ECR, EC2, S3, EFS, Amplify, Route53, LightSail, IAM, SNS, SES, SQS, API Gateway, DymanoDB, CloudWatch, Secret Manager, IAM, S3, DynamoDB, BitBucket, CloudFormation, API Gateway, EventBridge, Code Pipeline)
- Project/Product management methods and tooling: Agility, Github Projects, Gitlab, CollabNet, Jira;
- Documentation: Markdown, JSDoc, Chromatic, Storybook, Confluence;
- Marketing method and tools: RGPD, defining a client avatar, niche and offer, SEO, Google Analytics 4, Plausible, Axeptio;
- Visual design tooling: Figma, Adobe XD
- CMS: Shopify, Wordpress/Bedrock, WooCommerce, Strapi CMS
- Anytics and monitoring: pm2 services, Axiom

## SIDE PROJECTS

---

- nobour-eshop.com: e-commerce web application in MERN Stack, AWS (EC2, Amplify), Docker, Github Actions
- CRM application (crm.devedanos.com): PHP, API-Platform, React.js
- Air B n'B replica application (sym-5-bnb.devedanos.com): PHP, Symfony
- Todo-list and classrooms management applications: NestJS, ReactJS, MongoDB, PostgreSQL, GraphQL and REST APIs, AWS, TypeScript
- Auctions application: Node.js, React.js, Serverless Framework AWS, AWS Lambda

## HOBBIES AND INTERESTS

---

- Self-growth: leadership, communication, speed reading techniques;
- Entrepreneurship: e-commerce through Amazon FBA, affiliate marketing;;
- Web development: Agility, Extreme Programming, No Estimates, Continuous Delivery Pipelines, Typescript, React-Query, web 3.0, DApps, mentoring junior developers, Conferences (API-Platform Con, Devternity, JS World Conference and React World Conference...);
- Full remote work;
- Sports: artistic gymnastics, fitness, swimming;
- Benevolent activity: Since 2018 volunteer at an English speaking nonprofit, full-time volunteer from 2015 to 2017 as one-on-one teacher, public speaker;

WEEKANDGO.COM

SINCE OCTOBER 2021

### FREELANCE FULL-STACK DEVELOPER NEXT.JS NODE.JS

#### Introduction:

Weekandgo.com aims to become the reference in marketplaces for the sale and rental of recreational vehicles (motor homes, vans, etc.). Rapido, a French motorhome manufacturer based in Mayenne (France, 53) and among the leaders of the European market is at the origin of the project. The development of the application started in March 2021 and went into production in February 2022.

#### Context:

To carry out the developments, Rapido called upon the company Leb Communication, a company that has existed for more than 30 years and that today is developing its activity in web application development. In turn, Leb Communication called upon my services to reinforce its team of developers with a profile specialized in React.js, Node.js and Next.js, bringing lead on the technical aspects and prioritization of developments. Our team is made of 1 backend developer, an integrator and a UX designer, and me as full-stack developer.

The WeekandGo application is composed of a REST business API in API Platform (Symfony and PHP), a headless REST CMS in Strapi for the blog posts and a SPA in Next.js. The SPA is divided into 3 parts: one accessible to all visitors, one reserved for private sellers and another for professional sellers associated with Rapido. But the heart of the application lies in the filtering of ads.

#### Tasks:

- Development of features on the SPA such as the creation, edition, deletion and filtering of offers, the access securing/restriction to resources, the structure of the blog elements and the static generation or server-side rendering of its pages....
- Dependencies and Node versions maintenance,
- Quality: automation and development of regression tests (unit, visual, interactive, application), construction of a Continuous Delivery Pipelines
- Choice and overall technical design, technical consultant
- Driving force in the prioritization and simplification of functionalities
- Contribution of Agile practices for developments (automation of deployment, collaborative tools, common language to the team, uniform code syntax...)

#### Technologies:

Node.js, Next.js (next-auth), React.js (react-hook-form), Redux, Jotai, React Context, CSS3, HTML5, Material UI/MUI, Strapi CMS, REST, JWT, HTTP, Docker, bash, cache client SWR, PHP, API Platform, OVH, Vercel, Heroku, MySQL, PostgreSQL, CORS, React Hook Form, Open-Graph tags, Google Recaptcha v3, Google Tag Manager, Axeptio cookie consent, Storybook, Chromatic, Jest, Cypress, Playwright, testing-library

**FREELANCE FULL-STACK DEVELOPER NODEJS REACTJS NESTJS TYPESCRIPT**Introduction:

Product Feedback is a startup created in 2019 in Marseilles (13, France) that aims to revolutionise the management of customer feedback on a product. It is inspired by the notion.so application.

Product Feedback started from the observation that gathering and synthesising user feedback on a product was difficult and time-consuming: information not shared with the whole team, missing sheets or supporting documents... To simplify this management, Product Feedback is developing an online collaborative interface, gathering in one place customer responses to forms, product team members' reports on these responses, and final decisions based on the reports. It plans to add additional services such as excel spreadsheet integration and more...

In March, its team was composed of a CEO, a CMO based in Marseilles and a CTO working from the Czech Republic. I was directly working with the CTO.

Context:

Product Feedback proposed me to work for them as a freelance instead of passing technical tests, before hiring me. It was a full-remote position.

On the 6th day of trial, I had my first interview with the CEO. During this latter, I learned that Product Feedback team was actually looking for a senior developer to substitute its CTO. Hence, I worked 6 days for Product Feedback.

Tasks:

- Participation in the implementation of the application architecture, design of database and the backend entities
- Development of a GraphQL API as well as its queries and mutations
- Development of a User Interface as a Single Page Application, with its forms and reactive components
- Development of a GraphQL API, with its resolvers, services, 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

## FREELANCE FULL-STACK DEVELOPER SYMFONY EMBERJS

### Introduction:

The final client is a Dutch company that manufactures safety shoes. It has a CRM developed by Refactory Devops, a Dutch one-man company.

### Context:

Refactory Devops called on my services to update his client's CRM. I worked on the update of its backend and frontend. Refactory Devops takes care of the deployments in staging and production environments, and the code review. The follow-up of the tickets is done on Gitlab.

### Tasks:

- Update of the REST API, from Symfony 3 to 4 and adaptation of its services containers configuration
- Update of the Single Page Application from EmberJS 2 to 3, with its Octane syntax in templates, replacement of the ember classes by native JS classes in the components, controllers, helpers, models
- Development of descriptive unit and integration tests with PHPUnit for the REST API, and Qunit for the SPA

### Technologies:

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

## PHP DEVELOPER

### Introduction:

Valeo is a large international company, founded in France, and one of the leaders in the automotive equipment market. It is composed of 5 activities: thermal systems, visibility systems, propulsion systems, comfort systems and driving aids, and aftermarket activities. Each activity has a Methods and Tools team, which in turn has an Engineering Information System (EIS) team. The EIS team manages, among other things, the software licenses, thus allowing the engineers to have the necessary digital tools for their work (design, 3D modeling, virtual test benches).

### Context:

During my apprenticeship, I joined the EIS team of the thermal systems group. Directly under the supervision of my manager and as the only developer on the project, I was in charge of the maintenance of Licence Server Status, a custom PHP application allowing to monitor Valeo's servers all over the world in order to check the status of software licences. This application stores information in a database, and then displays it on a dashboard, providing alerts and sending emails when a license is about to expire. This project has gone through several versions over 15 years, and has often been rewritten from scratch by successive apprentices. The goal of this project is to help EIS in the prevention of renewal and purchase of software licenses to better manage the budgets of Valeo.

### Tasks:

- Suggestion and implementation of development practices, architecture, and technologies in order to preserve the project and not have to rewrite it anymore (using versioning, deploying programs via git instead of ftp, using a ticket management system, switching the project to OOP and MVC, adding automated tests and CI/CD...)
- Development and maintenance of the UI
- Development of charts in ChartJS
- Definition, development and maintenance of PHP scripts querying the servers and their software license daemons
- Designing of the architecture and implementation of the database
- Development of SQL queries
- Technology watching on Symfony 5 by developing an Air B n'B replica accessible on [sym-5-bnb.devedanos.com](http://sym-5-bnb.devedanos.com)
- Deployment in pre-production and production environments
- Development of Proof of Concepts for the custom PHP application License Server Status on Symfony 5 (not completed due to the COVID-19 pandemic)

### 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

## FREELANCE WEB DEVELOPER (JUNIOR)

### Introduction:

Refactory Devops is a Dutch one-man Digital Services Company. Its developer has more than 15 years of experience. It develops its own devops solutions, builds websites and applications for his clients.

### Context:

In 2019, I was planning to move to the Netherlands. For a few months I worked alongside the developer of Refactory Devops on projects for his clients, as well as his own startup project. I was working both from France and Nijmegen in Netherlands.

### Tasks:

- Implementation of the language internationalization of a Single Page Application
- Development of the authentication system via JWT and OAuth2
- Web integration for showcase sites via CMS
- Maintenance and correction of bugs on Single Page Applications
- Migration of websites from a web hosting service to another

### Technologies:

Neos Framework, Neos CMS, EmberJS 2, L18n, Javascript, HTML5, CSS3, PHP 7, YAML, cPanel;



Frontend development with EmberJS

### Introduction:

This project corresponds to the portfolio website of my micro enterprise for its activity related to webdevelopment. This website is available on [www.devedanos.com](http://www.devedanos.com).

It is a Single Page Application based on EmberJS, using the Server Side Rendering technique thanks to the ember-fastboot extension. But it is also able to use the static site generation through JAMStack thanks to the empress extension. I use Server Side Rendering to bring a better UX and a good SEO to my site. It is deployed on Netlify. The purpose of this site is to make visible my web development activity registered web under the trademark "Devedanos".

### Tasks:

- Organization of my tasks on NotionSO
- Preparation of the UI design: choice of colors, logo, drawing of components
- Registration of the trademark at the INPI
- Development of UI components and internationalization
- Development of features through TDD
- Deployment in production
- SEO (use of keywords, HTML semantic, deferred scripts)

### Technologies:

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

## Full-stack MERN development and Docker

### Introduction:

Personal project deployed on nobour-eshop.com and realized in nearly 400 hours of work.

In order to improve my skills on the MERN stack, I took the "Mern eCommerce from scratch" training on Udemy. As I felt that this training did not go as far as I wanted, nor fast enough, nor that the application produced was usable in real conditions, I added features and modified the structure of the project by myself. This application allows a merchant to put several products on sale and to be paid via Paypal. An emailing system sends emails to the store owner when the stock of products is low, when an order must be prepared. Emails are also sent to customers to confirm the creation of the account, the modification of the password, the confirmation of the purchase, the sending of the order. An invoice is automatically generated upon purchase and can be downloaded by both the merchant and the buyer. Only the shipping is not managed by the application.

### Tasks:

On the backend side, this training was a revision. On the frontend side, this training allowed me to :

- Take in hand Redux to manage the global state (store) of the SPA in React.js;
- Set up actions and reducers to execute HTTP CRUDS requests to the REST API, retrieve the product cart from the local storage;
- Set up access restriction to admin-only views and redirects using React Router hooks;

As for what I personally added to the project, or even learned on occasion on my own:

- Development of PDF invoice generation features, inventory management, order cancellation, emailing (account creation, password reset, order shipping, new order alert with the merchant and low stock);
- Docker image definition, commissioning and deployment of Docker containers in AWS ECR, EC2
- Implemented multi-stage CI/CD pipeline and production and staging environments via Github actions for the backend and CI/CD pipeline via AWS Amplify for the frontend;
- Securing the Express.js REST API with Express CORS and helmet middlewares
- Creation of an infrastructure for the application ecosystem on AWS
- Creation of databases for all environments on MongoDB Atlas services and Docker containers

### 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);

Backend development Cloud Serverless AWS Lambda NodeJS

#### Introduction:

In order to improve my skills on AWS and to make sure I understood how to use it after taking the course "AWS - The 12 fundamental services of the Cloud n°1", I took the training "Serverless Framework bootcamp : NodeJS, AWS & Microservices", both on Udemy. The latter allowed me to develop an auction application, provisioning resources on AWS with Serverless Framework. I developed 3 microservices: authentication, auction and notification services and connected them to a Single Page Application in React.js.

The source code is distributed on 3 github repositories:

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

#### Tasks:

- Development of HTTP endpoints performing CRUD via API Gateway, calling in turn Lambda functions
- Use DynamoDB as a DBMS to perform CRUD on data and create document schemas for data validation and error handling
- Implementation of message queues for communication between microservices via AWS Simple Queue Service
- Set up scheduled events via AWS EventBridge to close an auction after one hour
- Management of the entire microservices architecture via AWS CloudFormation indirectly exploited by the serverless.yaml files of the Serverless Framework (interconnection of services by retrieving their arn, url, environment...).
- Setting up object storage via AWS S3
- Implementation of email notifications via AWS SES to inform if the auction has been won by someone or has not found a buyer
- Implementation of an authentication and authorization middleware in JWT via API Gateway Lambda Authorizer, optimizing costs if the same JWT was used less than 300 seconds ago on the same API Gateway endpoint

#### Technologies:

Node.js, React.js, 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

Backend development NodeJS NestJS TypeScript

#### Introduction:

In order to learn more about NestJS and to make sure that my switch from PHP and Symfony to NodeJS was a good choice, I took a short online training on Udemy: "NestJS Zero to Hero - Modern TypeScript Back-end development".

It included two projects to create a todo list and a school class management system. I developed a REST API using PostgreSQL for the todo-list and a GraphQL API, its mutations and queries using MongoDB as DBMS for the school class management system

#### Tasks:

- Development of unit tests with Jest
- Design and development of REST and GraphQL APIs performing CRUD operations
- Authentication and Authorization for back-end applications,
- Development of scripts with TypeORM to interact with MongoDB and PostgreSQL databases
- Password hashing and storing sensitive information
- Deployment of the APIs at a production-ready state to Amazon Web Services
- Implementation of efficient logging in the backend applications
- Implementation of data validation and using Pipes
- Modelling entities for the persistence layer
- Development of Data Transfer Objects (DTO)

#### Technologies:

Node.js, TypeScript, NestJS, React.js, Apollo-server, PostgreSQL, git, Github, TypeORM, HTTP, JWT, Jest, SSL, MongoDB, Postman, PgAdmin, Jest;

## Full-stack development ReactJS API-Platform

### Introduction:

In order to know how to develop with React and create a REST API with PHP, I followed the "Modern development with ApiPlatform (Symfony) and React" training by Lior Chamla over one month. Through this course, I developed a Customer Relationship Management application, composed of a SinglePage Application in ReactJS and a REST API based on API-Platform, respecting the JSON API specification.

It is deployed on Heroku and accessible on [crm.devedanos.com](http://crm.devedanos.com).

### Tasks:

- Development of the SPA and forms
- Development and configuration of the JSON API
- Implementation of the database architecture
- Development of scripts and queries for filtering, scheduling, pagination
- Management of data serialization/deserialization
- Development of security: data validation, authentication by JWT
- Manual management of client-side data caching
- Deployment of the whole application on Heroku

### Technologies:

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

Full-stack development Javascript NodeJS

#### Introduction:

As part of my training for the title of Application Designer Developer at My Digital School Paris, I was in a team of 5 persons composed of a web designer, 2 web marketers, and another web developer. Our project allowed web designers and marketers of our team to validate their school year.

I was appointed project manager by the team and also had the role of lead developer. This allowed me to develop skills in marketing, web design, project management.

The aim of our "Worker Genius" project was to facilitate the recognition of the skills of digital professionals and their job search. To do so, we started the development of an application being at the same time a technical test platform, a job board and a real time exchange between recruiters and digital professionals. Concerning the development, I mainly focused on the backend, leaving the majority of the frontend development to the other developer of the team.

The application was composed of several HTML pages rendered by an ExpressJS server and a different REST API.

#### Tasks:

- Definition of technology choices, project structure
- Definition and development of the database model structure
- Development of a REST API connected to MongoDB, also validating JSON payloads
- Development of authentication with JWT
- Development of a websocket server for a chat service
- Development of a part of the UI
- Deployment on PlanetHoster

#### Technologies:

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