

SKILLS

BACK END DEVELOPMENT |

NodeJS, NextJS, Linux, REST APIs, ----- 7 Years

NextJS, Docker, ElectronJS, XHTTP, JSON-RPC, ----- 5 Years

Python3, Bash, **AWS**, EC2, Lambda, IAM, SQS, SNS, DynamoDB, APIG ---- 2 Years

FRONT END DEVELOPMENT |

ReactJS, React Native, JavaScript, CSS3, HTML5, ----- 7 Years

TypeScript, TailwindCSS ----- 2 Years

DATABASE | MySQL, Prisma, PlanetScale ----- 5 Years

LANGUAGES | ENGLISH, FRENCH, CHINESE (HSK4)

OFFICE | EXCEL, WORD, PPT, OpenOffice, WPS

DRIVING LICENSE | passed on June 29th 2023.

Full Stack Developer

Information

Age : 31

Nation : France

Name : Alexander

Surname : Leonetti

Marriage : Single

Contact

+33683038209

al.leo@wanadoo.fr

Education

Engineering Degree
of Microelectronics and
telecommunication
Polytech Marseille
2012-2016

EXPERIENCE

LARGE SCALE EMAIL CAMPAIGN on AWS Lambda & NODEJS ES6. [2 years]

- Built **email web scrapper** in order to harvest **tens of thousands** of customers emails addresses and automated the marketing process using **Browser JS Plugin** Async JavaScript and React.
- Front end : in the browser the plugin was extracting any emails from the loaded page source code whatever its **URL** may be.
- Plugin communications to SMTP servers were all done using **xhttp requests**.
- Processed data on **nodejs serverless Lambda on AWS** and a **DynamoDB** database on AWS to store collected mails.
- All processes and **API** communication were done with **Async ES6 and AWS API Gateway**.
- The mail sender part was done using **SMTPjs**. and using **axios**.
- Later **sending template mails** functionality has been added to the browser plugin itself making it possible to send mail campaign directly from the browser, just after the collecting phase.
- Later on a **standalone version** of this system has been requested by the customer, it has been built using **electron.js** framework.

2018 - 2023 : Self-Employed at 葱蜚尼(EnFeiNi)
a subcontractor software company.

CLOUD TRADING SYSTEM with AWS, EC2, Redis, NodeJS. [3+ years]

- Ultra Fast response time with **Redis RAM-database on AWS**.
- **Server** location and specification chosen to have **lowest latency** from Binance servers.
- Server is hosted on **Amazon EC2** platform, running headless Ubuntu Server.
- Designed **Autonomous trading server** using **REST APIs** and **NodeJS**.
- Data was streamed from exchange platform to local server. Then **middleware** took care of detecting price signals and initiated trades.
- Each trades have been generated automatically according to the chosen strategy.
- Each trades have been executed on the online exchange with **Async ES6**.

**2017 - Military Interface for Nuclear Submarine dashboard
using Full Stack NodeJS - Websocket - Javascript
Employed at Astek for Naval Group [1 year]**

- Created a **NodeJS** server communicating via **Websocket**.
- Developed **full stack NodeJS-Javascript** kiosk interface for military submarine's **touch screens** and navigation.
- Designed a **Nodejs server** that synchronized multiple touch screens according to operator's interactions.
- **MULTI SCREEN** communication was made possible with **WebSockets** .
- **Hosting** was provided on a local **Linux Ubuntu** machine and later on **Docker**.
- Developing was conducted on **VIM** and team cooperation was structured via **SCRUM**
- Developing the entire front end interface was built with vanilla **javascript**.

2018 - 2023 : Employed at 恩斐尼(EnFeiNi):

**REACT TypeScript, React Native and NextJS
(converted dozens of pages/design into applications and/or websites)**

- Converted customer designs into functional **React** web pages.
- Converted customer designs into **NextJS** (full websites with **APIs**)
- Converted customer designs into **React Native apps**.
- I still maintain some of this projects to this day.

DogePay / LitePay QR Code Payment System, Just like Alipay.

- Created a DogeCoin/LiteCoin **blockchain** payment server.
- The code base is compatible with both **Dogecoin** and **Litecoin** networks.
- Used **Bitcore-lib-doge** in order to create random private keys and wallets.
- Public and **private keys** are stored on a **mongodb** database for each users.
- Used **ExpressJS**, **Cors** and **Axios** modules to manage **API** server requests.
- Switched from **JSON-RPC** wallet communication to using dogechain APIs in order to achieve on chain balance inquiries. The main reason being that having a blockchain explorer locally cost a lot of resources.
- Front end : payment pages are generated using **EJS engine** for simplicity.
- **Unit testing** of API end points : made using **Chai and Mocha** nodejs modules
- Debugging has been done using **curl** and **postman** which i frequently use.
- This project will be deployed directly on **linux nodejs servers** or on cloud.

Other Projects :

PYTHON

- Back tested and simulated trend following strategies using **Python3**. in various cryptocurrency, gold and equity markets, and on different time frames.
- Fundamental Analysis was conducted strictly by human, but the technical and **data analysis** was conducted using **python** programming. simple strategies such as moving averages, grid trading and other price spikes system were used.
- The combination of all these factors helped the traders reach up to **500%** yearly return for 3 consecutive years.

Mobile App Educative Game “Flash Anzan”. in Next JS

- Built a **full stack NextJS** website ranking hundreds of Flash Anzan players.
- Originally the idea was built on a **PHP back end** with a classic **MYSQL database** and **ReactJS front End**.
- Being a fan of **NodeJS** and blazing fast execution, i felt obliged to rewrite the whole project for nodejs and then **NextJS**.
- **NextJS** generates html pages directly from server side. it is basically server side Reactjs.
- The database is hosted on **Planetscale a MySQL cloud database** provider.
- It is worth noting that NextJS cannot use databases directly, and i had to implement the use of the **Prisma library**.
- Authentication has never been more simple, since any user can connect instantly with google and github account, thanks to the **NextAuth library**.
- Main Challenge was to build an **ultra fast interface**, as top Flash Anzan performers, can achieve up to 10 calculations per seconds. There is no room for any type of delays.

R

- Simulated trading strategies with R language on all sorts of markets.
- Simulated investment returns on different time frames.
- Generated entry and exit trading signals.
- Studied the statistics associated with all these markets.

Proxy Plugin

- Re-Designed interface for VPN and proxy browser firefox plugin, using vanilla **Javascript** and custom **CSS3** sheets for a customer.

MediaBlock Plugin

- Developed a webpage scrapper that removes mainstream social media feeds and addictive loops, such as **youtube recommendation and infinite scrolls**.
- I'm still maintaining the project, according to Social Media code base changes.

Japanese Abbacus for Fast mental Arithmetics.

- Developed a **React.js** Japanese Abbacus for high speed mental calculation training.
- The main challenge of the project was that it needed ultra fast response time from the touch screen, to the tune of **50ms**, as abacus users can perform up to 10 calculations per seconds.

Linux

- Administered and debugged many **Linux** Systems and networks.
- I like to install and test **Linux Distributions** on old or small hardware as a hobby.
- My Work environment, is a **Lubuntu** distribution, with **Qtile window manager**, and