



Florent CARREZ - Développeur Backend Go/Python (3/4 ans exp.)

- Sociable
- Prévoyant
- Curieux

Pau / Télétravail / Hybride (≥ 40% télétravail)

florent.carrez.fc@gmail.com

06.01.43.02.32

Retrouvez mon CV en ligne: <https://cv.carrez.dev>

Expériences & Formation

Upfluence, Développeur Backend Go
06/2024 → 09/2025 (~1 an)

Conception, implémentation et déploiement (avec GitHub et Kubernetes) de nouvelles fonctionnalités pour une API REST. Correction de bug et intervention sur la production pour débloquer le support.

Micro Service, Thrift (protocole de communication semblable à gRPC), Redis, RabbitMQ, Postegres et SQLite (pour les tests)

Findit (Stedis), Développeur Go/Python
03/2023 → 03/2024 (~ 1an)

Migration du code Python vers du code Go, implémentation de nouvelle fonctionnalité pour leur API REST, structuration des données pour les rendre plus facilement exploitable. Correction de bug, conseil technique sur le Go et dockerisation de leur déploiement sur AWS.

Go, Python, MongoDB, AWS, Docker

Owlint, Développeur Backend Python
11/2023 → 02/2024 (4 mois)

Implémentation de fonctionnalité pour leur API REST autour de l'OSINT

Python, MongoDB, EventSourcing, Micro service

SII - Airbus, Ingénieur logiciels
12/2022 → 11/2023 (~1 an)

- Implémentation d'un logiciel from scratch en Python avec Pandas, OpenPyXL et PyQt.

- R&D interne à Airbus en C++17/Qt sur de la prévision de collision au décollage/au sol/à l'atterrissage.

Diplôme d'Ingénieur Informatique : EPITA

diplômé en 2021

Ecole d'ingénieur spécialisé en informatique

Bac+5

Compétences

Go ★★★★★

Python ★★★★★

C++ (17+) ★★★★★

Typescript ★★★★★

Swift, V, Rust ★★★★★

Framework JS : Svelte, Vu.JS

DevOps : Docker, K8s

Protocoles/Framework : REST, GraphQL, gRPC, Thrift, Microservice

Base de donnée : Postgres, SQLite, MongoDB

Niveau technique (lu/parlé/écrit)

Mes approches professionnelles

Grande autonomie
Autodidacte
Bienveillant
Clean code
Réactif

Loisirs

Karaté (ceinture noire) • Bivouac • Japonais (langue et culture) • Cinéma / Films / Animés

Unigorme : Go - En cours

ORM Go dont le but est d'être adaptable pour tous les Système de Gestion de Base de Données. Une première ébauche est disponible mais il reste encore plein d'aspect que je souhaite améliorer pour le rendre plus facile à utiliser, plus volatile et aussi plus riche en fonctionnalités

Potappli : Go, Svelte, PostgreSQL, SQLite, K8s - En cours

Projet de site web — mobile first — ayant pour but de faciliter la gestion de son potager en sauvegardant les données des plantes (tels que "quand semer/récolter", "quel type de sol", ...). Permettra également d'aider à organiser sa parcelle, de stocker les recettes utilisant nos plantes et proposer des recettes selon la saison. Le Backend est fait en Go, pour l'instant la base de donnée est en SQLite pour plus de simplicité mais je le développe de manière à pouvoir facilement passer sur Postgres. SQLite sera toujours utiliser pour les tests. Pour l'instant j'utilise le framework *Fiber* pour la gestion du serveur HTTP et exposer l'API REST. Côté Frontend, j'utilise SvelteKit (en TS) avec TailwindCss et je fais en sorte que l'interface soit *mobile-friendly*. Pour la partie DevOps, j'ai commencé à configurer un cluster K8s en local et une fois que ça sera bien rodé je compte déployer un cluster sur mon VPS.

Nu - mon langage de programmation : Go/Python, Notion- En cours

Je développe mon propre langage de programmation (pour le plaisir). Il s'agira d'un langage à part entière au même titre que le Go, le C++ ou le Python. J'ai terminé la conceptualisation de la plupart des fonctionnalités et de la syntaxe. L'analyse lexical (scanner) est terminée et je suis actuellement entrain de faire le parser (analyse grammatical -> génération de l'AST). Mon langage sera transpiler en C++ avant d'être compiler, mais je souhaite également faire en sorte qu'il soit interprétable.

Jusqu'à présent j'ai fait le code en Go car j'aime beaucoup ce langage, mais avec les nombreux projets que j'ai désormais en Go et en plus de plusieurs contraintes qui rallonge le développement du parser, j'ai décidé de faire une première version en Python (ce qui me permet globalement d'être plus pour le développement ; même si je perdrai du temps pour la partie compilation). Dès qu'une première version de mon langage sera disponible (avec le minimum nécessaire en terme d'implémentation du standard : read/write), je m'occuperais de réécrire le compilateur en Nu.