

# Projet Backend FastAPI

Ce document détaille le travail réalisé sur le projet backend encadré par Gabriele. Il présente les étapes clés du développement, les défis rencontrés et les apprentissages acquis.

# Présentation Générale du Projet

Le projet consistait en la création d'une API backend robuste et évolutive, développée en Python à l'aide du framework FastAPI. Cette API est conçue pour gérer diverses données métier essentielles, structurant l'information de manière logique et efficace.



## Python & FastAPI

Un choix technologique moderne pour la performance et la simplicité.



## Gestion de Données Métier

Consultants, entités, professions, sites et régions.



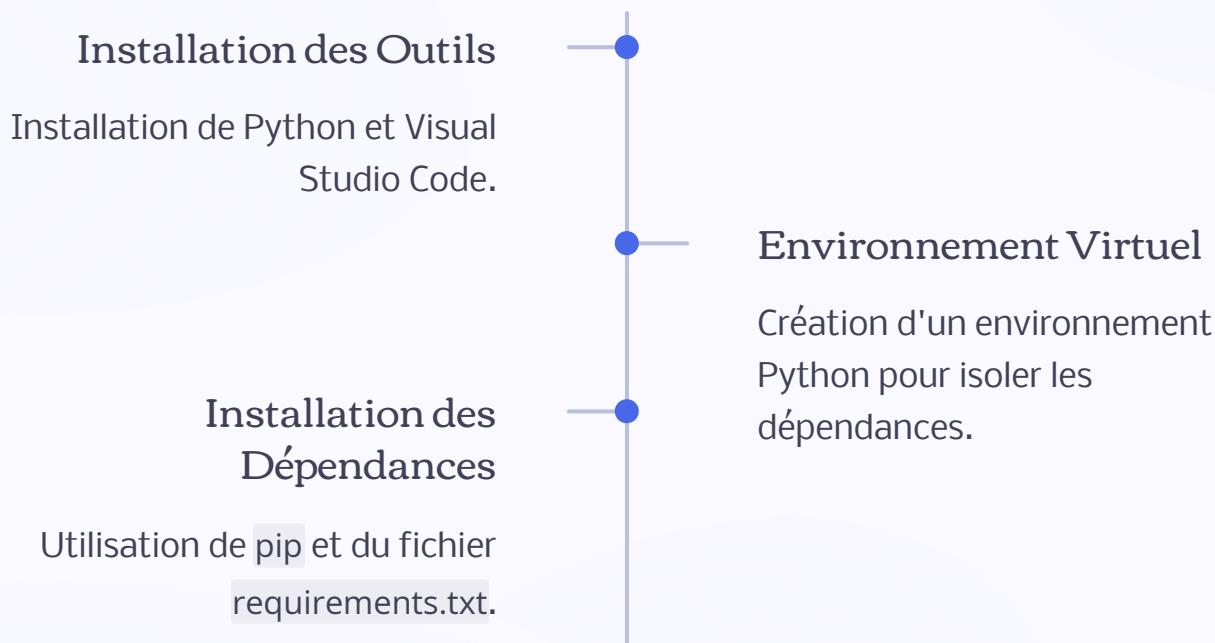
## MongoDB

Une base de données NoSQL flexible et performante.

L'architecture suit une séparation claire entre les contrôleurs (gestion des requêtes), les services (logique métier) et les modèles (structure des données et interaction avec la base de données), garantissant maintenabilité et scalabilité.

# Mise en Place de l'Environnement de Travail

La première phase cruciale du projet a été la configuration d'un environnement de développement optimal. Cette étape est fondamentale pour assurer une collaboration efficace et une exécution stable du projet.



Les dépendances installées incluent des librairies backend (FastAPI, Uvicorn), des outils de base de données (Beanie, Motor, Pydantic) et des outils de qualité de code (Bandit, Black, Flake8, Isort, Pylint, Radon, Coverage).

# Installation et Configuration de MongoDB

Un défi significatif a été l'adaptation de l'environnement de base de données. Initialement conçu pour Docker, des problèmes de compatibilité ont nécessité une approche alternative.

## Défi Docker

Difficultés techniques pour lancer MongoDB via Docker sur ma machine.

## Solution Locale

Configuration d'une instance MongoDB locale en remplacement.

## Adaptation de la Configuration

Modification de la chaîne de connexion pour pointer vers localhost.

## Validation

Tests approfondis pour garantir une connexion fonctionnelle et stable.

Cette adaptation a permis de surmonter les obstacles initiaux et d'assurer le bon démarrage du projet, prouvant la capacité à trouver des solutions créatives face aux contraintes techniques.

# Utilisation de Git et GitLab pour le Contrôle de Version

Parallèlement au développement technique, une immersion approfondie dans les outils de gestion de version, Git et GitLab, a été essentielle. Ces outils sont au cœur des pratiques de développement moderne en équipe.



## Prise en Main de Git

1

Apprentissage des commandes fondamentales: commit, branch, merge.



## Gestion de Dépôt avec GitLab

2

Utilisation de GitLab pour le suivi des modifications et la collaboration.



## Workflow d'Entreprise

3

Compréhension des merge requests et de la gestion de branches dédiées.

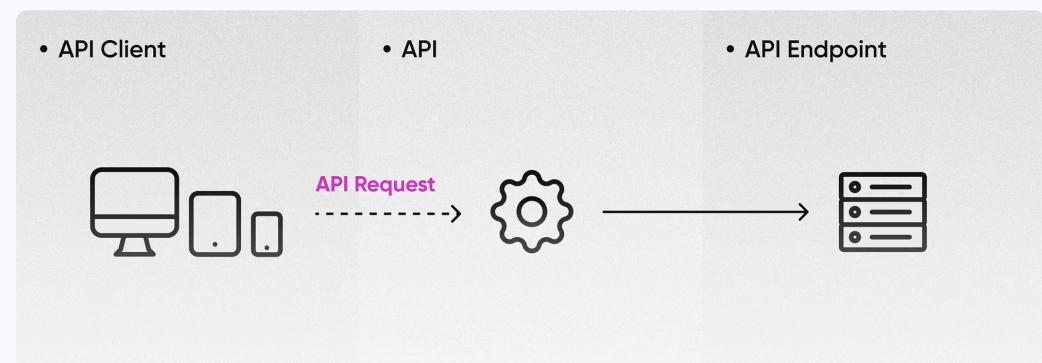
Cette expérience a permis d'adopter un workflow de développement organisé, reproductible et collaboratif, similaire à ceux utilisés dans les entreprises technologiques.

# Développement de l'API et Gestion des Entités Métier

Le cœur du projet a résidé dans le développement et l'amélioration de l'API, avec une attention particulière à l'intégration de nouvelles entités métier. Chaque entité a été traitée avec rigueur, en respectant l'architecture en couches.

## Structure d'une Entité

- **Contrôleurs (Routes API):** Définition des points d'accès HTTP (GET, POST, PUT, DELETE).
- **Services (Logique Métier):** Implémentation des règles et opérations spécifiques à chaque entité.
- **Modèles (DTO & MongoDB):** Définition de la structure des données et leur persistance dans MongoDB.



J'ai implémenté les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) pour structurer les données et permettre une manipulation efficace des informations des consultants, professions, sites et régions.

# Gestion des Relations entre les Entités

Un aspect fondamental du projet a été la modélisation et la gestion des relations complexes entre les différentes entités métier. Ces relations sont cruciales pour refléter la réalité des données d'entreprise.



Par exemple, un consultant peut être associé à plusieurs professions ou travailler sur divers sites, tandis qu'une profession peut être exercée par plusieurs consultants. Cette complexité justifie l'implémentation de relations de type "plusieurs à plusieurs" (N-N).

Ces choix de modélisation offrent une grande flexibilité et une meilleure représentation des cas d'usage réels, permettant une gestion plus dynamique et interconnectée des informations.

# Robustesse de l'API et Gestion des Erreurs

La qualité d'une API se mesure aussi à sa capacité à gérer les erreurs de manière élégante et informative. Un effort particulier a été mis sur la gestion précise des codes HTTP et des messages d'erreur.



## Code 404 : Ressource Introuvable

Indique que l'élément demandé n'existe pas, offrant une réponse claire à l'utilisateur.



## Code 409 : Conflit

Utilisé lorsque l'opération demandée crée un conflit (ex: données existantes), évitant les doublons.



## Code 400 : Mauvaise Requête

Signale une erreur dans la requête du client, avec des détails pour la corriger.

Cette approche permet de fournir une API plus compréhensible et plus facile à intégrer pour les clients, améliorant ainsi l'expérience développeur et la fiabilité globale du système.

# Leçons Apprises et Compétences Développées

Ce projet a été une opportunité inestimable d'acquérir une expérience pratique et de renforcer mes compétences techniques et professionnelles dans le développement backend.

- 1 Environnement de Dev  
Configuration et gestion complète d'un environnement de développement.
- 2 Gestion de Dépendances  
Maîtrise des outils pour la gestion des librairies et la base de données.
- 3 Git & GitLab  
Application des bonnes pratiques de contrôle de version et de collaboration.
- 4 Architecture Backend  
Compréhension approfondie d'une architecture API professionnelle.
- 5 Développement API  
Conception et implémentation d'endpoints robustes et sécurisés.
- 6 Gestion des Erreurs  
Mise en œuvre d'une gestion d'erreurs claire et informative.

Ces apprentissages sont directement applicables à des situations réelles en entreprise, consolidant une base solide pour ma carrière de développeur.

# Conclusion : Vers un Développement Professionnel

Ce projet a été bien plus qu'un simple exercice technique ; il a représenté une immersion complète dans les réalités du développement backend moderne. Les défis rencontrés, notamment avec la configuration de MongoDB et l'apprentissage de Git/GitLab, ont été des catalyseurs pour l'acquisition de compétences précieuses.

L'expérience de Gabriele a permis de renforcer ma compréhension de l'architecture API, de la gestion des données métier et de l'importance d'une gestion rigoureuse des erreurs. Je suis désormais mieux préparé à aborder des projets complexes et à contribuer efficacement au sein d'une équipe de développement.