

# Rapport du Projet

---

## Plateforme de profils pour freelances

### Introduction

Ce projet consiste à développer une plateforme de gestion des profils de freelances pour répondre à une problématique rencontrée par une startup :

Actuellement, les informations concernant les freelances sont échangées de manière non structurée (par Email ou WhatsApp), ce qui rend la gestion difficile et inefficace.

L'objectif est de proposer une solution technique moderne, structurée, simple à déployer et évolutive.

### Choix des technologies / logiciels

#### Pourquoi utiliser GraphQL ?

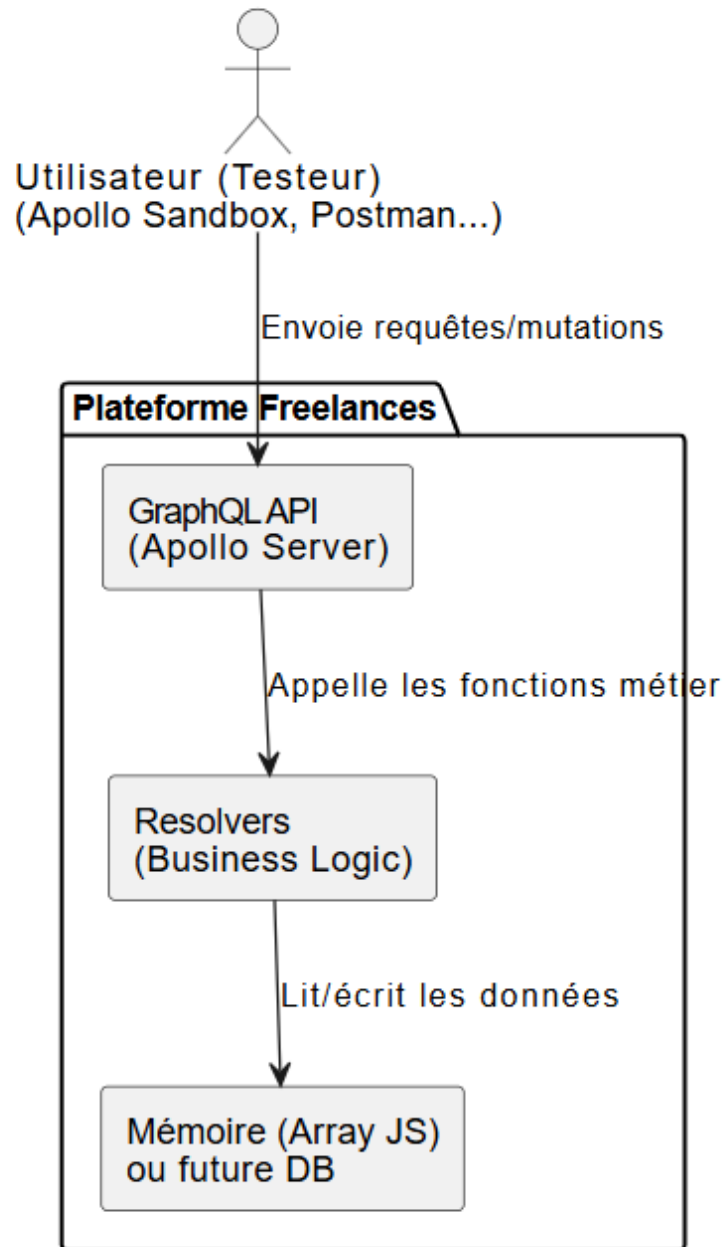
Nous avons choisi GraphQL pour plusieurs raisons :

- Permet d'avoir une API flexible → le client choisit les données qu'il veut.
- Moins de surconsommation de données qu'avec REST (pas d'overfetching ni d'underfetching).
- Facile à étendre → nous avons ajouté les notions de compétences, de liens professionnels et de communications.
- Compatible avec de nombreux clients : React, Angular, mobile...

#### Outils utilisés

- Node.js: Exécuter le serveur
- Apollo Server: Créer l'API GraphQL
- Apollo Sandbox: Tester l'API facilement
- Visual Studio Code (VSCode): Éditeur de code
- Git / GitHub: Sauvegarde et versioning du projet
- Draw.io: Créer les diagrammes (classes, architecture)

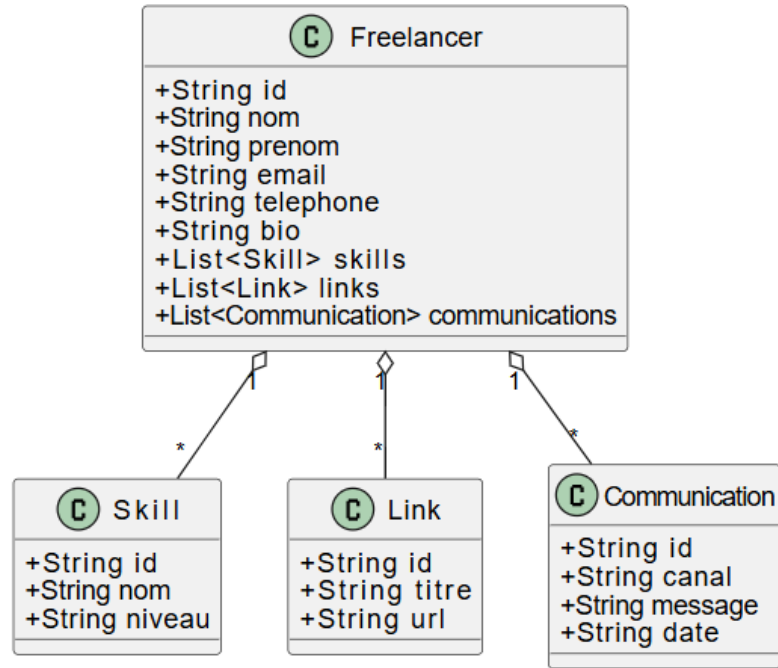
## Architecture du projet



Le projet est découpé de manière simple et modulaire :

- index.js → Point d'entrée du serveur GraphQL
- schema.graphql → Définition du schéma (types, queries, mutations)
- resolvers.js → Implémentation de la logique métier
- package.json → Gestion des dépendances Node.js

## Diagramme de classes



## Liste des fonctionnalités (sous forme de web services)

- Créer un profil freelance.
- Modifier un profil freelance.
- Supprimer un profil freelance.
- Ajouter une compétence.
- Supprimer une compétence.
- Ajouter un lien professionnel.
- Supprimer un lien professionnel.
- Ajouter une communication (email ou WhatsApp).
- Supprimer une communication.
- Lister tous les freelances.
- Consulter un freelance par ID.

## Étapes de travail (Comment travailler sur ce projet)

### Étape 1 : Initialisation

- Créer le dossier du projet
- Initialiser avec `npm init -y`
- Installer les dépendances : `npm install apollo-server graphql uuid`

### Étape 2 : Mise en place du serveur

- Créer `index.js` pour configurer Apollo Server.
- Créer le fichier `schema.graphql` pour définir le modèle de données et les mutations.
- Créer `resolvers.js` pour coder la logique métier.

### Étape 3 : Test

- Lancer le serveur : `npm start`
- Aller sur Apollo Sandbox pour tester les requêtes et mutations.

### Étape 4 : Extensions

- Ajout de la gestion des communications (Email / WhatsApp).
- Ajout de la suppression individuelle pour skills / links / communications.

### Étape 5 : Finalisation

- Écrire un fichier `README.md`.
- Générer le diagramme de classes sur [draw.io](https://draw.io).
- Préparer un rapport détaillé.

## Conclusion

Le projet de plateforme de profils pour freelances permet :

- Une gestion centralisée et structurée des profils.
- Une traçabilité des échanges (Email / WhatsApp).
- Une API flexible et extensible.
- Une architecture simple à maintenir.

Ce projet peut facilement évoluer vers :

- Une base de données persistante (MongoDB, PostgreSQL).
- Une interface web (React, Angular).
- Des fonctionnalités avancées (authentification, filtres, recherche).

## Résumé du projet

- Solution : API GraphQL
- Stockage : Données en mémoire
- Fonctionnalités : CRUD Freelances + gestion compétences, liens, communications
- Documentation claire
- Diagramme de classes fourni