

PROJET D'APPLICATION

Rapport Final

Projet 16: Extraction de données de l'API AlumnForce et enregistrement en base de données

Client: Centrale Nantes Alumni

Encadrants: Christelle ROUSSEAU, Stéphane BURET

Responsables Techniques: Larissa ALBUQUERQUE NOGUEIRA, Clara MATTOS MEDEIROS

Nantes
20 Décembre 2024

Sommaire

DÉFINITIONS	3
1. SYNTHÈSE DU PROJET	4
2. INTRODUCTION	5
2.1. Contexte	5
2.2. Objectifs	6
2.3. Portée du logiciel	6
3. MÉTHODOLOGIE	7
3.1. Méthodologie de développement	7
3.1.1 Tests unitaires	7
3.3. Outils et logiciels utilisés	8
3.4. Planification	9
4. RÉSULTATS	12
4.1. Rendus Finaux	12
4.1.1 Base de données	12
4.1.2 Scripts pour charger la base de données	14
4.1.3 Documentation	16
4.1. Démonstration de fonctionnement	16
4.1.1 Tables, type_utilisateur, mail et type	18
4.1.2 Tables adresse, ville et pays	18
4.1.3 Tables a_un_diplome, diplome, specialisation et ecole	19
4.2. Adhérence au planning initial	20
4.3. Estimation des coûts	20
5. DIFFICULTÉS RENCONTRÉS	21
6. SUITE DU PROJET	23
7. CONCLUSION	24
8. ANNEXES	25
8.1. Dépôt Github du Projet	25
8.2. Exemple d'information du POST /api/v2/customer/profile/people	25
8.3. Exemple d'information du POST /api/v2/customer/academic/member/	27
9. RÉFÉRENCES	28

DÉFINITIONS

API : Interface de Programmation d'Application.

CSV : Comma-separated values. Format de fichier de valeurs séparées par des virgules.

Alumnforce : Plateforme (réseau social) gérant les informations des étudiants et diplômés de Centrale Nantes

CNA: Centrale Nantes Alumni.

DSI: Direction des Systèmes d'Information de Centrale Nantes.

1. SYNTHÈSE DU PROJET

Ce projet s'inscrit dans le cadre de l'amélioration de la gestion des données des alumni de CNA, en vue de faciliter leur accès et leur mise à jour par le personnel concerné. L'objectif principal était de concevoir et de mettre en place une base de données locale, synchronisée avec la plateforme AlumnForce, tout en assurant une documentation technique claire et détaillée pour en garantir l'utilisation et la continuité.

Le projet a été développé en suivant la méthodologie du cycle en V, permettant une validation méthodique des étapes et des fonctionnalités. Les travaux ont inclus la conception d'un modèle relationnel, le développement de scripts pour la création et la population initiale de la base, ainsi que l'extraction et l'insertion des données via l'API de AlumnForce.

Les livrables principaux comprennent les scripts de gestion de la base de données, des programmes pour l'insertion et mise à jour des données, et une documentation complète décrivant l'architecture technique, les outils nécessaires et les étapes d'utilisation.

Le dépôt GitHub du projet peut être accédé sur le lien suivant: [GitHub](#).

2. INTRODUCTION

2.1. Contexte

Afin de connecter ses diplômés et étudiants, l'École Centrale de Nantes dispose d'une association appelée Centrale Nantes Alumni (CNA). Cette association fédère, rapproche et anime la communauté des centraliens de Nantes afin de promouvoir l'École et ses diplômés, de contribuer au rayonnement de la communauté des centraliens de Nantes et de développer des relations avec le monde économique au bénéfice de ses membres et de Centrale Nantes.

À partir de son site web, les membres ont accès aux profils des étudiants et diplômés, peuvent rejoindre des groupes géographiques, métiers, sportifs ou culturels créés sur la plateforme (avec des forums de discussions), consulter des actualités et des événements liés à Centrale Nantes notamment (comme la cérémonie de diplomation, des événements, conférences, afterworks, webinaires, visites d'entreprises...), et accéder à une boutique en ligne proposant des articles exclusifs aux couleurs de Centrale Nantes, tels que des T-shirts, pulls et mugs, entre autres.

Ce réseau est fourni par une entreprise tierce : MEVIA. Cette dernière propose la plateforme web AlumnForce avec ses fonctionnalités et est responsable du stockage des données des membres et de leurs informations personnelles, sur leur parcours académique, leur carrière et les groupes rejoints sur le site.

L'extraction des données depuis la plateforme AlumnForce n'est pas un processus direct. Elle nécessite une requête au support technique du prestataire pour obtenir les données, lesquelles sont transmises au format CSV. Par ailleurs, l'insertion des données est également complexe. L'équipe utilise actuellement un dépôt distant géré par le prestataire, où un fichier CSV contenant les informations des étudiants – extraites d'une autre plateforme d'inscription utilisée à Centrale – est transmis pour être intégré chaque jour à la base de données par le tiers entreprise.

Ce processus d'extraction, de mise à jour et d'ajout d'informations dans la base de données présente plusieurs inconvénients. Non seulement il est chronophage, mais il expose également l'équipe à des risques liés à la manipulation directe de données sensibles. Ces risques incluent des erreurs potentielles lors de la transmission et de l'intégration des informations dans la base de données.

2.2. Objectifs

Depuis un an, AlumnForce, le prestataire de stockage en base de données de CNA, a mis en place une API de service qui facilite l'interaction avec la base de données, que ce soit pour importation ou exportation. Le fait d'interroger l'existant grâce à l'API permet d'éviter d'écraser des cursus déjà suivis antérieurement par certains membres.

D'ailleurs, le projet a pour objectif général la mise en place d'une base de données locale qui exploitera les données depuis l'API de la plateforme AlumnForce, afin de permettre au personnel de CNA d'accéder plus facilement aux données des membres. Cette solution est accompagnée d'une documentation détaillée expliquant son utilisation, garantissant ainsi une adoption fluide et une continuité dans sa gestion.

Les rendus comprennent un script pour la création de la base de données, un autre pour sa population et sa mise à jour, ainsi qu'une documentation technique détaillée.

La documentation inclut la description de l'architecture de la base de données et du programme développé, ainsi que les logiciels et leurs versions nécessaires pour accéder aux données et un tutoriel pour déployer les scripts.

2.3. Portée du logiciel

Le projet a pour objectif principal d'améliorer la gestion des données des alumni en récupérant l'ensemble des informations déjà enregistrées sur la plateforme AlumnForce et en les intégrant dans une base de données locale, spécifiquement conçue pour répondre aux besoins du personnel de CNA. Cette nouvelle base de données est accompagnée d'une documentation technique détaillée, permettant une utilisation efficace et durable par l'équipe en charge.

Il est important de préciser que le projet se limite à la récupération des données existantes et à leur intégration dans une base de données locale. Bien que l'équipe de développement ait envisagé d'inclure le développement d'un système permettant l'insertion de nouvelles données dans cette base, la complexité du projet s'est avérée plus importante que prévu. Par conséquent, la création d'un outil de visualisation des données et le développement d'un système pour l'insertion de nouvelles informations ont été exclus du périmètre défini pour ce projet. Ces fonctionnalités supplémentaires pourraient néanmoins être considérées dans le cadre d'une phase ultérieure.

3. MÉTHODOLOGIE

3.1. Méthodologie de développement

La méthodologie adoptée pour ce projet a été le cycle en V. À chaque fonctionnalité ajoutée au code, l'équipe a mis en place des tests afin de s'assurer que les spécifications initialement définies étaient respectées et alignées avec les besoins réels du projet, tout en validant la phase précédente.

Le choix du cycle en V pour ce projet s'explique par la clarté des exigences dès le départ, la nécessité de livrer des résultats fiables et robustes, et les contraintes de temps et de ressources.

Contrairement aux méthodes agiles, qui favorisent l'adaptabilité mais peuvent entraîner des délais et des ajustements fréquents, le cycle en V a permis une planification rigoureuse et une validation méthodique à chaque étape.

Cette approche s'est avérée particulièrement adaptée à une équipe avec une expérience limitée, réduisant les risques liés aux malentendus et assurant une correspondance stricte entre les spécifications initiales et les livrables, comme la base de données, les scripts et la documentation technique.

3.1.1 Tests unitaires

Les tests réalisés au cours du projet ont joué un rôle essentiel dans la validation de chaque étape du développement. Cependant, la nature complexe et interdépendante du projet a imposé certaines contraintes qui ont influencé notre approche en matière de tests.

À mesure que le projet avançait, nous avons effectué des tests manuels réguliers pour valider l'insertion des données dans la base de données. Cette approche a consisté à exécuter les scripts d'insertion, puis à interroger la base via l'interface pgAdmin pour vérifier que :

- Les données extraites de l'API étaient correctement insérées dans les tables correspondantes.
- Les relations entre les entités (clés étrangères) étaient maintenues et cohérentes.
- Aucune duplication ou anomalie n'était présente dans les enregistrements.

Ces tests étaient particulièrement utiles pour valider des fonctionnalités critiques, telles que la gestion des relations complexes entre utilisateurs, adresses, et diplômes. Bien qu'efficace

pour détecter des erreurs, cette méthode est toutefois limitée en termes d'automatisation et de reproductibilité.

Le projet repose fortement sur des interactions avec des systèmes externes (l'API AlumnForce et la base PostgreSQL), ce qui complique la création de tests unitaires pour l'ensemble des fonctionnalités. Voici les principaux défis rencontrés :

- Les scripts dépendent des données dynamiques extraites via l'API. Simuler ces données de manière réaliste nécessiterait la création de nombreux cas de test pour couvrir toutes les variations possibles.
- L'interaction avec la base de données implique un état mutable, rendant difficile l'isolement des tests.
- L'utilisation de requêtes asynchrones pour extraire les données ajoute une couche de complexité. Tester ces fonctionnalités nécessiterait des outils spécifiques pour simuler et synchroniser les réponses de l'API.
- Les scripts insèrent des données dans plusieurs tables interconnectées. Tester chaque fonction individuellement sans affecter les autres requiert une configuration et une gestion minutieuses des données de test.

Malgré ces limitations, nous avons développé un fichier de tests unitaires, concentré sur les fonctions de formatage des données, qui jouent un rôle clé dans la préparation des entrées avant leur insertion dans la base. Ce fichier, *test_format.py*, utilise le module unittest de Python pour valider les bons comportements des fonctions. Ces tests ont permis d'identifier des erreurs dans les premières versions des fonctions, comme des cas de caractères non pris en charge ou des chaînes non correctement échappées.

3.3. Outils et logiciels utilisés

Le projet a utilisé plusieurs outils et technologies pour assurer la gestion efficace de la base de données, le développement du logiciel et la communication entre les membres de l'équipe.

Pour la gestion et la conception de la base de données, le principal logiciel utilisé est pgAdmin 4, associé à PostgreSQL 16.4, qui est une base de données relationnelle open-source robuste et fiable, permettant de gérer et manipuler les données de manière efficace. En

complément, l'outil SQL Power Architect 1.0.9 a été utilisé pour la conception de schémas de bases de données, facilitant la création et la gestion des modèles relationnels.

Le langage de programmation choisi pour le développement du projet est Python 3, en raison de sa flexibilité et de ses bibliothèques puissantes, qui facilitent la récupération de données via une API et la manipulation des bases de données. Les principales bibliothèques utilisées sont :

- asyncio et aiohttp : pour la gestion des requêtes asynchrones à l'API.
- psycopg2 : pour interagir avec la base de données PostgreSQL.
- unicode : pour normaliser les chaînes de caractères.
- json : pour manipuler les données en format JSON.
- Visual Studio Code a été utilisé comme l'environnement de développement intégré (IDE).

Pour le suivi du code et la gestion des versions, Git a été utilisé en combinaison avec GitHub. Le lien du dépôt GitHub était partagé entre les parties prenantes du projet, qui pourraient suivre le projet à distance.

Concernant la gestion de projet et communication interne, Google Drive a été utilisé pour le stockage et le partage de fichiers, notamment des documents Word et des rapports. Pour le suivi des tâches et l'organisation du travail de l'équipe, l'outil Trello était utilisé. Il s'agit d'une application web de gestion de projet visuel basée sur le principe du Kanban, permettant de suivre l'avancement des tâches et d'assigner des responsabilités.

La communication entre les membres de l'équipe s'est faite principalement par email et parfois via l'application Slack, un outil de messagerie instantanée.

3.4. Planification

Le projet était initialement divisé en trois phases. Au fur et à mesure que le projet avançait, l'équipe a ressenti le besoin de diviser la phase 2 en deux pour pouvoir atteindre un objectif faisable et concret pour la finalisation du projet. C'est pourquoi les phases qui seront détaillées ci-dessous diffèrent légèrement des phases prévues sur le cahier des charges.

Phase 1 - Initial - Familiarisation - Découverte de l'API

Dans un premier temps, l'objectif principal est de se familiariser avec l'API d'AlumnForce, de connaître la syntaxe pour réaliser des extractions et de réussir à visualiser sur le terminal les champs de données envisagées. Le livrable de cette phase est un script dans la

langue de programmation choisie qui se connecte à l'API et sort les informations de quelques alumni dans le terminal. Toutes les informations pertinentes pour l'utilisation de l'API seront ajoutées à la documentation du projet.

Phase 2 - Modélisation et création de la base de données

Dans la deuxième phase, l'objectif est de concevoir, de modéliser et de créer la base de données qui récoltera les informations sorties d'AlumnForce. Pour cela, les livrables de cette phase sont le modèle conceptuel de données et le modèle physique de données, ainsi qu'un script pour la création des tableaux. En plus, cette phase comportera une documentation très importante pour la continuation du projet, responsable pour expliquer le fonctionnement de la base de données, sa conception et comment la remplir.

Phase 3 - Final - Insertion des données et mise à jour de la base de données

Dans la troisième et dernière phase, l'objectif est de remplir la base de données locale avec les informations récupérées via l'API d'AlumnForce et de mettre en place un mécanisme de mise à jour automatique de ces données.

Le livrable de cette phase est un programme développé en Python, conçu pour effectuer l'extraction et la mise à jour des données de manière efficace. Ce programme garantit que la base locale reste continuellement synchronisée avec la base de données d'AlumnForce, offrant ainsi une solution fiable et automatisée pour maintenir l'intégrité des informations.

Phase 4 - Application pour faciliter l'utilisateur à accéder aux infos de l'API

Dans le cas idéal, la finalisation de ce projet sera accompagnée de la création d'une application qui donne facilement aux infos de AlumnForce via API sans utilisation des scripts par le personnel de CNA. Cette application aurait une documentation claire et extensive et permettrait la continuation du projet vers la création d'une base de données partagé entre les différents secteurs de l'école, et pas seulement locale dans l'ordinateur de qui souhaite l'utiliser.

Cependant, comme discuté avec les encadrants du projet, il est improbable que cette phase arrive à terme faute de temps pour le développement du projet. Donc, cette phase sera classifiée comme en 'stand-by' pour qu'un groupe prochain puisse prendre le projet en main et la continuer.

Phase 1: Découverte de l'API	Deadline
Réunion de suivi	03 Octobre
Prise en main de l'API	03 Octobre
Réalisation des tests avec l'API	18 Octobre
Réunion de suivi	17 Octobre
Livrable : Cahier des charges	11 Octobre
Livrable : Script pour accéder les données	18 Octobre

Phase 2 - Modélisation et création de la base de données	Deadline
Tester les manières d'extraire une quantité importante de données via l'API	20 Octobre
Conceptionner une base de données pour garder les informations	20 Octobre
Livrable : Diagrammes UML / MCD / MPD / documents de conception	25 Octobre
Livrable : Script pour la création de la base de données	25 Octobre

Phase 3 - Insertion des données et mise à jour de la base de données	Deadline
Insérer et/ou mettre à jour les données sur les tables qui sont indépendantes	29 Novembre
Insérer et/ou mettre à jour les données sur les tables qui dépendent fortement des autres	20 Décembre
Livrable : Squelette de rapport final	06 Décembre
Livrable : Script pour l'insertion des données sur la base de données	06 Décembre

Phase 4 - Application pour faciliter l'utilisateur à accéder aux infos de l'API	Deadline
Étude - possibilité de faire une interface visuel	20 Décembre
Livrable: Rapport final	20 Décembre

4. RÉSULTATS

4.1. Rendus Finaux

Sur le plan technique, les rendus permettant la création de la base de données et sa population sont disponibles sur le dépôt GitHub partagé avec les encadrants. Ces livrables témoignent d'une progression constante dans le projet.

4.1.1 Base de données

La base de données constitue un élément central du projet, conçue en PostgreSQL pour assurer une structure robuste et évolutive. Elle est le fruit d'une modélisation soignée, respectant les normes des bases relationnelles et intégrant des relations entre les entités clés telles que utilisateur, diplôme, adresse, école et spécialisation. Ce modèle a été optimisé pour répondre aux besoins spécifiques de gestion des données de l'association Centrale Nantes Alumni.

Le script de création, disponible sur le dépôt GitHub, permet de générer automatiquement la base de données. Il contient des instructions SQL pour définir :

- Les tables, leurs colonnes, et leurs types de données.
- Les contraintes d'intégrité telles que les clés primaires, les clés étrangères et les contraintes d'unicité.
- Les séquences pour générer des identifiants uniques.

L'une des particularités de cette base est la table pays, qui est pré-remplie avec une liste de pays. Cette table est fixe et ne dépend pas des données extraites via l'API AlumnForce. En revanche, toutes les autres tables (telles que personne, adresse, ou diplôme) sont peuplées dynamiquement à partir des données extraites.

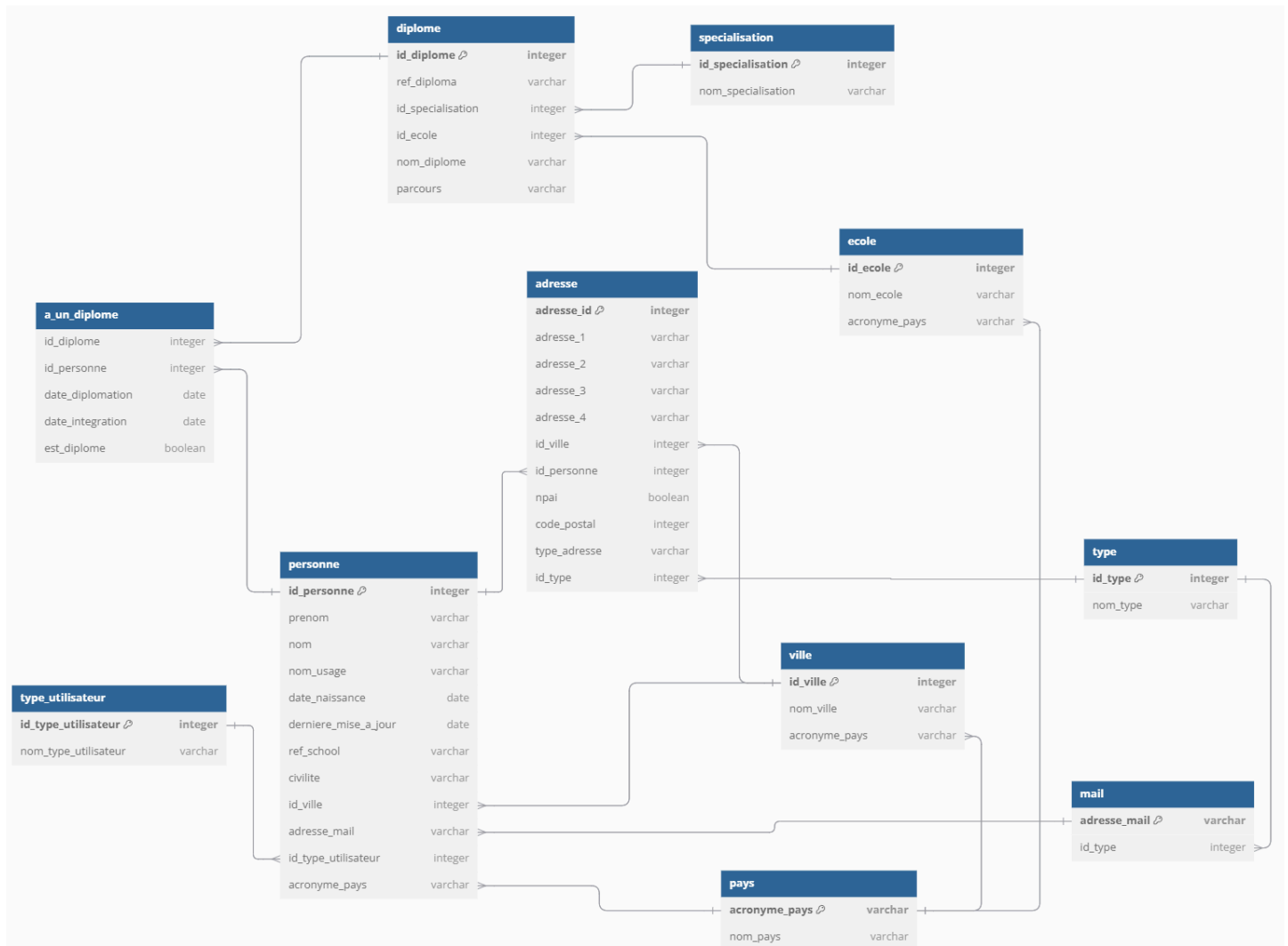


Figure 1 - Modèle Physique des Données

La modélisation des données vise également à assurer l'intégrité et la cohérence des informations insérées dans la base. Les valeurs discrètes possibles, telles que les types d'utilisateur, d'adresse ou d'adresse mail, sont clairement définies dans des tables spécifiques. Cette structure permet une identification rapide des anomalies dans les données et une validation automatique des informations avant leur insertion, empêchant ainsi l'introduction de données incorrectes ou non conformes.

Un point particulier concerne la table pays, qui est pré-remplie avec une liste de pays et considérée comme fixe. Toutes les autres tables (telles que personne, adresse, ou diplôme) sont dynamiquement peuplées à partir des données extraites via l'API AlumnForce. Si l'API évolue

pour inclure de nouvelles données ou structures, ces tables peuvent être automatiquement mises à jour sans modification supplémentaire du script.

4.1.2 Scripts pour charger la base de données

Les scripts développés en Python exploitent les fonctionnalités de l'API AlumnForce pour récupérer les données nécessaires, les traiter et les insérer dans la base de données. Voici un aperçu des étapes principales et des scripts utilisés :

1. Connexion à la base de données

Le script **databasePush.py** initialise une connexion avec la base de données PostgreSQL, permettant l'exécution des requêtes SQL.

2. Récupération des données via l'API

Le projet a impliqué l'extraction de données pour environ 25000 Alumni, générant un grand trafic de requêtes depuis l'API. Une approche initiale basée sur des requêtes synchrones s'est révélée inefficace, nécessitant plus de 30 minutes pour récupérer toutes les informations.

Afin d'optimiser ce processus, le groupe a étudié et implémenté une solution asynchrone. Un sémaphore a été utilisé pour effectuer jusqu'à 50 requêtes asynchrones simultanément, chaque requête récupérant une page contenant 100 utilisateurs. Une fois toutes les requêtes terminées, les résultats sont combinés pour former un dictionnaire consolidé contenant les données extraites.

Cette approche a permis de réduire considérablement le temps de récupération des données à 8 minutes tout en respectant les limites de l'API. Le script de récupération des données peut être trouvé sur le GitHub de l'équipe comme **fetchData.py**.

3. Insertion des données

Chaque table dispose d'une fonction dédiée pour insérer/mettre à jour ses enregistrements. La documentation détaillée de chaque fonction utilisée par le groupe dans ce projet est présentée sur GitHub.

4. Validation et intégrité des données

Chaque insertion est précédée de vérifications pour s'assurer que les enregistrements ne sont pas dupliqués.

Les scripts appliquent également une logique stricte pour valider les valeurs discrètes (types d'utilisateur, d'adresse, ou de mail), réduisant ainsi les risques d'anomalies ou d'incohérences dans les données.

5. Rapport sur les performances

Le script mesure la durée d'exécution pour chaque étape, offrant une visibilité sur les performances de l'insertion des données.

En conclusion, ces scripts offrent une solution fiable et automatisée pour synchroniser les données de la plateforme AlumnForce avec la base de données locale. Ils garantissent une mise à jour continue et une intégrité optimale des informations, répondant ainsi aux besoins du personnel de CNA.

L'interaction entre les différents fichiers peut être modélisée par un diagramme fonctionnel montré sur la Figure 2.

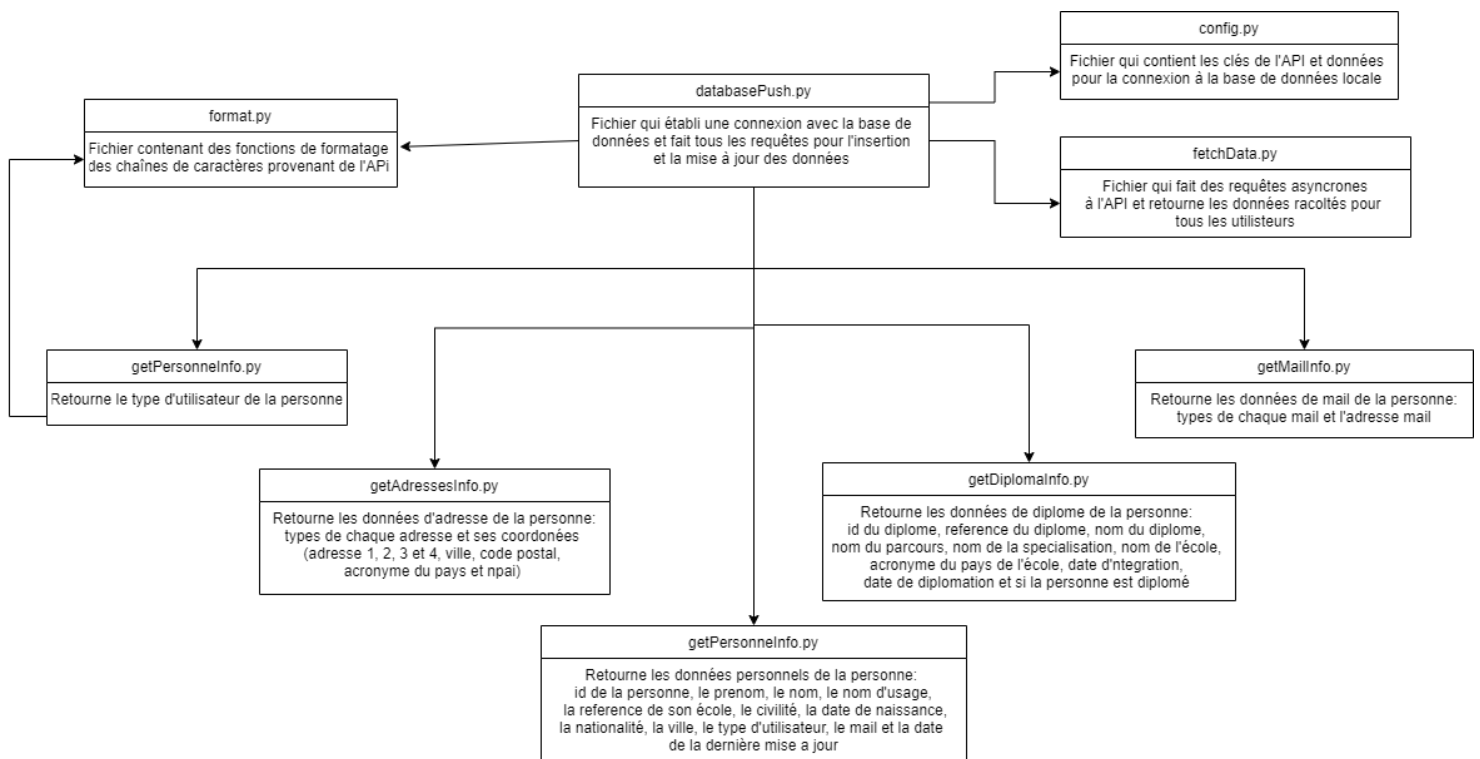


Figure 2 - Diagramme Fonctionnel

4.1.3 Documentation

L'ensemble de la documentation du projet est réuni dans le [dépôt GitHub du projet](#). Il s'agit de l'archive README, que c'est un tutoriel détaillé sur la façon d'utiliser et d'exécuter le script disponible sur le dépôt, d'une documentation au format HTML qui décrit chaque fonction du code, ainsi que des spécifications et des rapports du projet.

Sur la première page, l'archive README décrit pas à pas comment utiliser les ressources développées pour le projet, en commençant par l'installation des bibliothèques nécessaires à l'exécution du code, la création de la base de données sur pgAdmin, le paramétrage des fichiers contenant l'accès à l'API et à la base de données, ainsi que l'exécution du script pour l'ajout des données dans la base de données créée localement.

Cette archive fournit également l'information sur l'emplacement où est sauvegardée la documentation du code, avec le code commenté, ainsi que les rapports et spécifications du projet.

4.1. Démonstration de fonctionnement

Pour illustrer le fonctionnement de la base de données et des scripts développés, nous avons inclus dans les annexes de ce rapport des exemples réels de données extraites via l'API AlumnForce, au format JSON. Chaque alumni nécessite deux requêtes distinctes à l'API :

- Une requête pour récupérer le profil utilisateur, incluant des informations personnelles comme les données de civilité (nom, prénom, date de naissance ...) et informations de contact (adresse postale et adresse mail).
- Une requête pour récupérer les diplômes associés, comprenant des détails tels que les spécialisations, les écoles et les années de diplomation.

Ces exemples figurent dans les annexes avec les champs pertinents à extraire mis en évidence en rouge. Ils permettent une comparaison directe entre les données issues de l'API et celles insérées dans la base de données une fois les scripts exécutés.

Pour démontrer le fonctionnement des scripts, nous avons suivi les instructions détaillées dans le fichier README.md, disponible sur le dépôt GitHub du projet. Voici les étapes principales :

1. Initialisation et exécution :

Les scripts de configuration et de récupération des données sont exécutés conformément aux étapes décrites dans le README. La procédure se termine par l'exécution du script principal, databasePush.py, qui insère les données extraites dans la base PostgreSQL locale.

2. Vérification des résultats :

À l'issue de l'exécution du script, le terminal affiche un résumé des actions effectuées, tel qu'illustré dans la Figure 3, qui détaille le nombre d'enregistrements ajoutés ou mis à jour pour chaque table.

```

PS D:\Info SI\test\projet-application\code> python databasePush.py
Chargement des données depuis l'API...
(Cette opération peut prendre quelques minutes)

Durée de récupération des données de l'API : 0:01:42.393124

-----

Insertion des données dans la base de données...

Établissement de la connexion à la base de données...
Connexion à la base de données établi.

Ajout des types d'utilisateur à la base de données...
Succès à l'ajout des types d'utilisateur à la base de données.

Ajout des types de mail à la base de données...
Succès à l'ajout des types de mail à la base de données.

Ajout des types d'adresse à la base de données...
Succès à l'ajout des types d'adresse à la base de données.

Ajout des specialisations à la base de données...
Succès à l'ajout des specialisations à la base de données.

Ajout ou mise à jour des données de mail à la base de données...
Succès à l'ajout ou mise à jour des mails à la base de données.

Ajout des villes à la base de données...
Succès à l'ajout des villes à la base de données.

Ajout ou mise à jour des personnes dans la base de données...
Succès de l'ajout ou mise à jour des personnes dans la base de données.

Ajout ou mise à jour des adresses à la base de données...
Succès à l'ajout ou mise à jour des adresses à la base de données.

Ajout des écoles à la base de données...
Succès de l'ajout des écoles à la base de données.

Ajout des diplômes à la base de données...
Succès à l'ajout des diplômes à la base de données.

Ajout ou mise à jour des liens diplômes/personnes à la base de données...
Succès à l'ajout ou mise à jour des liens diplômes/personnes dans la base de données.

-----

Durée de la mise à jour de la base de données : 0:03:13.524106
Durée totale du script : 0:04:55.964097

```

Figure 3 - Terminal après exécution de databasePush.py

3. Validation dans la base de données :

Une fois les scripts exécutés, nous avons utilisé l'interface pgAdmin pour effectuer des requêtes SQL sur la base de données. Par exemple, nous avons vérifié les informations ajoutées

pour l'alumni ayant l'ID 23699, en croisant ses données avec celles extraites initialement de l'API.

Cette procédure démontre non seulement l'efficacité des scripts pour récupérer et insérer les données, mais également l'intégrité des informations stockées dans la base. Les résultats obtenus confirment que la base de données est synchronisée avec les données de l'API et prête à répondre aux besoins de gestion des alumni de Centrale Nantes Alumni.

4.1.1 Tables, type_utilisateur, mail et type

```
SELECT id_personne,
       prenom,
       nom,
       date_naissance,
       derniere_mise_a_jour,
       civilite,
       adresse_mail,
       nom_type,
       nom_type_utilisateur
FROM   personne
NATURAL JOIN mail
NATURAL JOIN type_utilisateur
NATURAL JOIN TYPE
WHERE  id_personne = 23699
```

	id_personne integer	prenom character varying	nom character varying	date_naissance date	derniere_mise_a_jour date	civilite character varying	adresse_mail character varying (128)	nom_type character varying	nom_type_utilisateur character varying (64)
1	23699	Clara	MATTOS MEDEIROS	2003-05-22	2024-04-29	Mme	mattosclara22@gmail.com	primary	Etudiant

4.1.2 Tables adresse, ville et pays

```
SELECT personne.id_personne,
       prenom,
       nom,
       adresse_1,
       adresse_2,
       adresse_3,
       adresse_4,
       code_postal,
```

```

        nom_ville,
        nom_pays,
        npai,
        type_adresse
FROM personne
JOIN adresse ON (personne.id_personne = adresse.id_personne)
JOIN ville ON (ville.id_ville = adresse.id_ville)
JOIN pays ON (pays.acronyme_pays = ville.acronyme_pays)
WHERE personne.id_personne = 23699

```

	id_personne integer	prenom character	nom character varying	adresse_1 character varying (128)	adresse_2 character varying (128)	adresse_3 character varying (128)	adresse_4 character varying (128)	code_postal integer	nom_ville character varying	nom_pays character varying	npai boolean	type_adresse character varying
1	23699	Clara	MATTOS MEDEIROS	Avenida Lucio Costa 3360		Bl 5 Ap 406		22630	Rio De Janeiro	Brésil	false	secondary

4.1.3 Tables a_un_diplome, diplome, specialisation et ecole

```

SELECT id_personne,
       prenom,
       nom,
       nom_diplome,
       nom_ecole,
       nom_pays,
       ref_diploma,
       id_specialisation,
       parcours,
       date_integregation,
       date_diplomation,
       est_diplome
FROM personne
NATURAL JOIN a_un_diplome
NATURAL JOIN diplome
JOIN ecole ON ecole.id_ecole = diplome.id_ecole
JOIN pays ON ecole.acronyme_pays = pays.acronyme_pays
WHERE id_personne = 23699

```

	id_personne integer	prenom character	nom character varying	nom_diplome character varying (256)	nom_ecole character varying (128)
1	23699	Clara	MATTOS MEDEIROS	Universidade Estadual De Campinas Ingenieur Reseau Time 2026 Br Double Diplome	Universidade Estadual De Campinas
2	23699	Clara	MATTOS MEDEIROS	Centrale Nantes Ingenieur Informatique Pour Les Systemes D'Information 2026 Fr Ingenieur	Centrale Nantes

nom_pays character varying (128)	ref_diploma character varying (128)	id_specialisation integer	parcours character varying (128)	date_integration date	date_diplomation date	est_diplome boolean
Brésil	DD_BRCAMPINA01_UNIVERSIDADEESTADUALDECAM	[null]	Double Diplome	2023-09-18	2026-12-05	false
France	PROG_INGE_OPTION_INGE_EL_INFOSI	120	Ingenieur	2023-09-18	2026-12-05	false

4.2. Adhérence au planning initial

Au cours du développement du projet, l'équipe a suivi le plan initial tout en s'adaptant aux défis rencontrés pour garantir une livraison conforme aux objectifs principaux. Bien que la complexité liée à l'extraction et au traitement des données se soit révélée importante, les solutions mises en place ont permis de respecter les délais impartis et de fournir des livrables de qualité.

Les données présentes dans AlumnForce, récupérées au format JSON avec des structures imbriquées, ont nécessité plusieurs requêtes pour extraire toutes les informations nécessaires. Pour répondre à ce défi, l'équipe a adopté une stratégie de population progressive de la base de données, associée à des vérifications systématiques pour éviter les doublons et garantir une insertion cohérente et fiable.

Bien que le projet ait atteint ses objectifs principaux, il est important de noter qu'il n'a pas été possible de développer une interface visuelle pour accéder à la base de données. Cette fonctionnalité, toujours considérée comme optionnelle dans le cadre du projet, aurait requis un temps de développement supplémentaire non compatible avec les contraintes du planning. L'équipe a donc choisi de concentrer ses efforts sur la création d'une base de données robuste et sur l'automatisation des scripts d'insertion, qui constituent les priorités définies dès le départ.

En conclusion, l'équipe a démontré sa capacité à surmonter les contraintes techniques et organisationnelles tout en livrant un système complet et opérationnel. Les résultats obtenus offrent une solution fiable et durable, tout en posant les bases pour des développements futurs, tels que l'ajout éventuel d'une interface utilisateur ou d'autres améliorations.

4.3. Estimation des coûts

Afin d'estimer le coût du projet, l'équipe a consigné les heures travaillées chaque semaine. À cet effet, il a été pris en compte que 70 % du coût du projet est consacré aux salaires des développeurs, 20 % à la maintenance de l'entreprise concernant les développeurs, et 10 % au paiement des logiciels utilisés.

Le tableau 1 présente le nombre d'heures travaillées par semaine par l'équipe pendant toute la durée du projet. La cinquième semaine correspondait à une période de vacances.

En considérant une équivalence de 8 heures pour une journée-homme, le travail réalisé représente approximativement 10 journées-homme.

Pour le calcul des salaires des développeurs, un taux horaire brut de 20 euros a été retenu pour des développeurs débutants (4). Le coût total du développement s'élève donc à 1 640 euros.

En ajoutant les 30 % liés à la maintenance et aux logiciels, le coût total du projet est estimé à environ 2 342,86 euros net.

Heures faites par semaine	
Semaine 1	03:30:00
Semaine 2	08:30:00
Semaine 3	08:30:00
Semaine 4	03:30:00
Semaine 5	00:00:00
Semaine 6	05:00:00
Semaine 7	09:30:00
Semaine 8	04:50:00
Semaine 9	03:20:00
Semaine 10	07:15:00
Semaine 11	05:30:00
Semaine 12	22:30:00
Total	81:55:00
Coût du projet (net)	€2.342,86

Tableau 1 - Nombre d'heures travaillés sur le projet

5. DIFFICULTÉS RENCONTRÉS

L'équipe a fait face à des difficultés sur différents axes pendant l'exécution du projet. Sur l'axe de développement et de conception, il y avait des difficultés concernant la base de données, la gestion de projet et les attentes définies pour le projet.

En ce qui concerne la définition du modèle physique de données, bien que le modèle initial semblait adéquat, plusieurs fautes ont été remarquées au fur et à mesure que les données étaient ajoutées à la base de données. Par exemple, la taille des champs des lignes d'adresse postale devait être augmentée, et certains champs initialement pensés comme non nuls se sont avérés nuls dans la base d'AlumnForce, entre autres.

Une autre difficulté résidait dans le maintien de l'intégrité des données lors de l'ajout des informations de ville et d'adresse à la base de données. Ce sont des champs saisis par les utilisateurs via l'API, et nous avons donc plusieurs entrées correspondant à un même résultat. Par exemple, la ville Sucé-sur-Erdre pouvait apparaître écrite en majuscules, avec ou sans accents, ou avec ou sans traits d'union. Cela augmentait la difficulté de vérifier si la donnée était déjà ajoutée ou non.

Pour les données d'adresse, l'utilisateur dispose de 4 champs pour les saisir, sans contraintes de taille de caractères ni de géolocalisation par un outil GPS. Il n'y avait donc pas de règle claire pour ces champs, ce qui rendait difficile leur traitement et la vérification de l'existence d'une adresse dans la base lors de l'ajout d'une nouvelle donnée. Cette problématique persiste dans la base de données, et la décision prise fut de ne pas les traiter et d'ajouter une nouvelle adresse seulement si elle n'existait pas déjà dans le champ principal.

En ce qui concerne les spécifications et la planification du projet, l'équipe a sous-estimé la complexité de l'extraction des données et de leur ajout à la base, peut-être en raison d'un manque d'expérience sur le sujet. Ainsi, l'équipe avait initialement prévu de développer une interface visuelle comme phase finale, mais la récupération des données, leur extraction, leur traitement et la communication avec l'API, suivis de leur ajout à la base de données locale, ont pris beaucoup de temps. Cela a empêché l'ajout, dans le délai imparti, de certaines données demandées par le client, comme les informations sur la participation à des groupes de Centrale et sur les carrières. Cependant, ces limitations ont été communiquées aux parties prenantes du projet, qui ont validé les ajustements de périmètre.

En ce qui concerne la gestion de projet, le manque d'expérience des développeurs a pu ralentir un peu la progression initiale. Malgré une bonne communication, des différences dans les approches logiques de programmation ont parfois conduit à des incohérences dans le code, principalement en début de projet. Cependant, avec une meilleure intégration et une compréhension des méthodes de travail de chacun, ces problèmes ont diminué. Encore, pour y remédier, l'équipe a adopté l'utilisation de Trello pour le suivi des activités et une répartition optimisée des tâches, et le renforcement de la communication par l'envoi hebdomadaire d'audios détaillés expliquant les avancées, les fonctionnalités développées et les prochaines étapes, renforçant ainsi la compréhension mutuelle et la coordination.

Bien que ces problèmes aient posé des défis, ils ont également été des opportunités d'apprentissage. En les analysant régulièrement et en mettant en place des actions correctives, l'équipe a pu transformer ces obstacles en éléments constructifs pour le développement du projet et pour éviter leur répétition à l'avenir.

6. SUITE DU PROJET

Il existe plusieurs axes d'amélioration et d'ajout pour le projet déjà réalisé.

Tout d'abord, on peut mentionner la suite logique du travail, qui consiste à finaliser l'insertion de toutes les données demandées par le client dans la base de données, telles que les informations sur le parcours de carrière des alumni, leurs fonctions associatives, leurs centres d'intérêt, etc.

Dans la même optique, une amélioration possible serait de garantir la synchronisation des informations de l'API AlumnForce avec la base de données, en implémentant la suppression automatique des données présentes dans la base mais absentes sur l'API.

Un autre axe d'amélioration serait de développer une application dotée d'une interface permettant aux membres de CNA d'ajouter facilement et instantanément des données à la base AlumnForce, via l'API par des requêtes de type POST.

En parallèle, il serait pertinent de concevoir une interface visuelle de Business Intelligence permettant d'exploiter les données des diplômés. Cette interface pourrait fournir des indicateurs tels que le nombre de diplômés par spécialisation chaque année, le nombre de doubles diplômes, les échanges internationaux, ou encore les élèves inscrits en master.

Par ailleurs, il reste essentiel de corriger les éventuels bugs et d'exécuter des tests fonctionnels exhaustifs sur la base de données. Malgré les tests réalisés, certaines incohérences subsistent, comme le fait que seuls les e-mails de type "principal" sont liés à une personne, tandis que d'autres types d'e-mails sont ajoutés sans être directement associés à un individu.

Enfin, concernant les tests unitaires, des pistes d'amélioration incluent l'utilisation de bibliothèques de simulation (mocking) pour simuler les réponses de l'API, la création d'une base de données temporaire pour isoler les tests d'insertion et valider les relations entre les tables, ainsi que l'automatisation des tests manuels via des scripts capables de comparer automatiquement les résultats extraits de l'API avec les données insérées dans la base.

En somme, les pistes d'amélioration pour ce travail initial sont infinies, laissant entrevoir un potentiel considérable pour l'évolution future du projet.

7. CONCLUSION

Ce projet a permis de répondre pleinement à l'objectif principal de faciliter la gestion des données des Alumni de Centrale Nantes grâce à la mise en place d'une base de données locale alimentée par les données extraites de la plateforme AlumnForce. Malgré les défis rencontrés, notamment la complexité imprévue liée à l'extraction et au traitement des données, l'équipe a su s'adapter en ajustant le modèle initial et en optimisant les processus de développement.

Toutes les étapes définies ont été menées à bien, incluant la création de la base de données, l'élaboration des scripts d'automatisation pour l'insertion et la mise à jour des données, ainsi que la production d'une documentation technique complète. Cette dernière garantit que la solution est non seulement opérationnelle mais également pérenne, facilitant ainsi son adoption et sa maintenance par l'association.

Nous exprimons notre gratitude à Mme Christelle Rousseau et à M. Stéphane Buret pour leurs orientations précieuses, leur disponibilité et leur soutien constant tout au long du projet. Leurs contributions ont été essentielles à la qualité des résultats obtenus et à l'atteinte des objectifs.

En conclusion, le projet a été un succès, offrant à l'association une solution robuste et fiable qui simplifie la gestion des données des Alumni, tout en posant les bases pour des développements futurs tels que l'ajout d'une interface utilisateur ou l'intégration de nouvelles fonctionnalités.

8. ANNEXES

8.1. Dépôt Github du Projet

GitHub du Projet: <https://github.com/PAPPL-Clari/projet-application>

- Ce répertoire contient les codes sources, un tutoriel de comment déployer le projet, les spécifications fonctionnelles du projet et les rapports de suivi.

8.2. Exemple d'information du POST **/api/v2/customer/profile/people**

```
{
  "id": "23699",
  "school_ref": "230628A",
  "hash": "XRm7G",
  "username": "mattosclara22@gmail.com",
  "display_name": "Clara MATTOS MEDEIROS",
  "is_contributor": false,
  "has_subscribed_newsletters": true,
  "job_title": "",
  "points": 0,
  "validation_date": "0000-00-00 00:00:00",
  "created": "2023-09-19 03:04:09",
  "logincount": "21",
  "updated": "2024-04-29 20:42:56",
  "last_activity": "2024-12-12 14:29:04",
  "date_end_access": "0000-00-00 00:00:00",
  "web_url":
    "https://centraliens-nantes.org/person/clara-mattos-medeiros/23699",
  "profile_completion_rate": 24,
  "_links": {
    "self": {
      "href": "/api/v2/customer/profile/people/23699"
    },
    "diplomas": {
      "href": "/api/v2/customer/academic/member/23699"
    },
    "career": {
      "href": "/api/v2/customer/career/member/23699/experience/"
    }
  },
  "embedded": {
    "emails": [
      {
        "address": "mattosclara22@gmail.com",
        "type": "primary",
        "is_username": true,
        "is_notification": true,
        "is_email_for_life": false,
        "is_verified": true,
        "is_blacklisted": false
      },
      {
        "address": "clara.mattos-medeiros@eleves.ec-nantes.fr",
        "type": "secondary",
        "is_username": false,
        "is_notification": false,
        "is_email_for_life": false,
        "is_verified": true,
        "is_blacklisted": false
      }
    ],
    "address": [
      {
        "type": "primary",
        "address": "Avenida Lúcio Costa 3360",
        "address_2": "",
        "address_3": "bl 5 ap 406",
        "address_4": "",
        "city": "0",
        "zip": "22630010",
        "sorting_code": "",
        "region": "",
        "administrative_area_iso": "0",
        "country_iso": "BR",
        "country_name": "Brésil",
        "npai": false,
        "landline": "",
        "mobile_prefix": "+33",
        "mobile": "07 44 81 53 52",
        "latitude": "-23.010890000000",
        "longitude": "-43.329810000000"
      },
      {
        "type": "secondary",
        "address": "Avenida Lúcio Costa 3360",
        "address_3": "bl 5 ap 406",
        "city": "RIO DE JANEIRO",
        "zip": "22630010",
        "region": "Rio de Janeiro",
        "country_iso": "BR",
        "country_name": "Brésil",
        "npai": false
      }
    ],
    "civil": {
      "id": "23699",
      "firstname": "Clara",
      "lastname": "MATTOS MEDEIROS",

```

```

"birthname": "MATTOS MEDEIROS",
"birthdate": "2003-05-22",
"name_used": "lastname",
"is_dead": false,
"gender": "0",
"title_of_honor": "",
"_embedded": {
  "civility": {
    "id": "2",
    "name": "Mme"
  },
  "marital_status": {
    "id": "0",
    "name": "Non renseigné"
  },
  "nationality": {
    "id": "6",
    "name": "Brésil"
  }
},
"avatar": {
  "user_id": "23699",
  "display_name": "Clara MATTOS MEDEIROS",
  "color": "#84de18",
  "initials": "CM",
  "_links": {
    "self": {
      "href": "/api/v2/customer/profile/people/23699"
    }
  }
},
"type": {
  "id": "5",
  "name": "Etudiant",
  "label": {
    "fr": "Etudiant",
    "en": "Student"
  },
  "type": "core",
  "default_role_id": "7",
  "is_assignable": false
},
"roles": [
  {
    "id": "5",
    "name": "Cotisant",
    "description": "usually: has done a contribution",
    "type": "core"
  },
  {
    "id": "7",
    "name": "Etudiant",
    "description": "etudiant",
    "type": "core"
  },
  {
    "id": "17",
    "name": "Abonné",
    "description": "subscriber",
    "type": "core"
  }
],

```

```

{
  "id": "18",
  "name": "Administrateur Contenu",
  "description": "Custom role: Administrateur Contenu",
  "type": "custom"
},
"activation_status": {
  "id": "3",
  "name": "Compte activé",
  "activated_at": "2023-11-11 00:49:05",
  "profile_count": 0
},
"navigation_language": {
  "id": "48",
  "code": "fr",
  "name": "français",
  "translations": {
    "default": "Français",
    "fr": "Français",
    "en": "French"
  }
},
"preference": {
  "created": "2023-11-11 00:49:05",
  "updated": "2023-11-11 00:49:05",
  "_embedded": {
    "notification": {
      "receive_school_emails": false,
      "receive_party_and_forum": true,
      "receive_newsletter": true,
      "receive_directory": true,
      "seeking_internship": true
    },
    "confidential": {
      "appear_in_directory": true,
      "hide_email_perso": true,
      "hide_email_pro": true,
      "hide_fax": true,
      "hide_fax_pro": true,
      "hide_phone": true,
      "hide_phone_pro": true,
      "hide_mobile": true,
      "hide_mobile_pro": true,
      "hide_address": true,
      "hide_address_pro": true,
      "hide_education": true
    },
    "participation": {
      "support_student": true,
      "life_of_network": true,
      "jury": false
    }
  }
},
"latest_diploma": {
  "id": "5153080",
  "school": "Universidade Estadual de Campinas",
  "reference": "DD_BRCAMPINA01_UNIVERSIDADEESTADUALDECA
M",
  "study_mode": "Continu",

```

```

    "active": false,
    "study_year": "BR",
    "graduation": "2026-12-05",
    "integration": "2023-09-18",
    "created": "0000-00-00 00:00:00",
    "updated": "2024-12-19 03:02:16",
    "is_graduated": false,
    "is_delegate": false,
    "entity": "Ingénieur Réseau Time",
    "specialization": "",
    "concentration": "Double Diplôme",
    "user_id": "23699",
    "diploma_ref":
"DD_BRCAMPINA01_UNIVERSIDADEESTADUALDECA
M",
    "full_name": "Universidade Estadual de Campinas -
Ingénieur Réseau Time - 2026 - BR - Double Diplôme",
    "_links": {
      "diploma": {
        "href":
"/api/v2/customer/academic/diploma/DD_BRCAMPINA01_UN
IVERSIDADEESTADUALDECAM"
      }
    }
  },
  "edit": {
    "href":
"/api/v2/customer/academic/member/23699/diploma/5153080"
  },
  "professional_status": {
    "id": 0,
    "name": "STATUS_NO_REQUEST",
    "label": "Sans attente professionnelle"
  },
  "terms_of_use": {
    "accepted": true,
    "accepted_at": "2023-11-11 00:49:03"
  }
}

```

8.3. Exemple d'information du POST **/api/v2/customer/academic/member/**

```

{
  "id": "23699",
  "id_school": "230628A",
  "_links": {
    "self": {
      "href": "/api/v2/customer/academic/member/23699"
    }
  },
  "embedded": {
    "diplomas": [
      {
        "id": "5153080",
        "school": "Universidade Estadual de Campinas",
        "reference":
"DD_BRCAMPINA01_UNIVERSIDADEESTADUALDE
CAM",
        "study_mode": "Continu",
        "active": false,
        "study_year": "BR",
        "graduation": "2026-12-05",
        "integration": "2023-09-18",
        "created": "0000-00-00 00:00:00",
        "updated": "2024-12-19 03:02:16",
        "is_graduated": false,
        "is_delegate": false,
        "entity": "Ingénieur Réseau Time",
        "specialization": "",
        "concentration": "Double Diplôme",
        "user_id": "23699",
        "diploma_ref":
"DD_BRCAMPINA01_UNIVERSIDADEESTADUALDE
CAM",
        "full_name": "Universidade Estadual de Campinas -
Ingénieur Réseau Time - 2026 - BR - Double Diplôme",
        "_links": {
          "diploma": {
            "href":
"/api/v2/customer/academic/diploma/DD_BRCAMPINA01
_UNIVERSIDADEESTADUALDECAM"
          },
          "edit": {
            "href":
"/api/v2/customer/academic/member/23699/diploma/51530
80"
          }
        }
      },
      {
        "id": "5153081",
        "school": "Centrale Nantes",
        "reference":
"PROG_INGE_OPTION_INGE_EI_INFOSI",
        "study_mode": "Continu",
        "active": false,
        "study_year": "FR",
        "graduation": "2026-12-05",
        "integration": "2023-09-18",
        "created": "0000-00-00 00:00:00",
        "updated": "2024-12-19 03:02:16",
        "is_graduated": false,
        "is_delegate": false,
        "entity": "Ingénieur",
        "specialization": "Informatique pour les Systèmes
d'Information",
        "concentration": "Ingénieur",
        "user_id": "23699",
        "diploma_ref":
"PROG_INGE_OPTION_INGE_EI_INFOSI",
        "full_name": "Centrale Nantes - Ingénieur -
Informatique pour les Systèmes d'Information - 2026 - FR -
Ingénieur",
        "_links": {

```

```

        "diploma": {
            "href":
"/api/v2/customer/academic/diploma/PROG_INGE_OPTI
ON_INGE_EI_INFOSI"
        },
        "edit": {
            "href":
"/api/v2/customer/academic/member/23699/diploma/51530
81"

```

9. RÉFÉRENCES

- (1) [Centrale Nantes Alumni: C'est quoi un Alumni?](#)
- (2) [Page web Centrale Nantes Alumni](#)
- (3) [Documentation de l'API de AlumnForce](#)
- (4) [Fiche métier : Ingénieure / Ingénieur informatique](#)