

JOBMARKET

Ce projet a pour but de mettre en avant vos compétences de Data Engineer. Vous allez regrouper des informations sur les offres d’emplois et les compagnies qui les proposent. À la fin du projet, vous aurez une meilleure vision du marché de l’emploi : quels secteurs recrutent le plus, quelles compétences sont requises, quelles sont les villes les plus actives etc ...

Récupération des données : Vue d'ensemble

Pour ce projet, nous avons exploré plusieurs sources d’offres d’emploi en ligne afin de centraliser les données et d'en tirer un maximum d'informations pertinentes.

Sources sélectionnées

1. API France Travail

- Très structurée, pas de redirection sur d’autres sources.
- Permet d’accéder à plusieurs API et pas seulement les offres d’emploi : API compétences, API métier en vogue, API régions dynamiques...

2. API The Muse

- Il y’a des redirections dans les API, mais des redirection interne (sur le site the Muse). Donc La structure est homogène entre les pages, ce qui rend le scraping plus accessible et automatisable.

3. API Adzuna

- Redirections externes fréquentes (vers LinkedIn ou autres), rendant le scraping plus difficile.

Structure des données normalisées

Les données sont normalisées selon le modèle suivant:

Champ	Description
id	Identifiant unique de l'offre
source	Source de données ('adzuna' ou 'france_travail')
title	Titre de l'offre
description	Description de l'offre
company_name	Nom de l'entreprise
location_name	Nom de la localisation
latitude	Latitude géographique
longitude	Longitude géographique

Champ	Description
date_created	Date de création de l'offre
date_updated	Date de mise à jour de l'offre
contract_type	Type de contrat (CDI, CDD, etc.)
contract_duration	Durée du contrat
working_hours	Temps de travail (Temps plein, Temps partiel)
salary_min	Salaire minimum
salary_max	Salaire maximum
salary_currency	Devise du salaire
salary_period	Période du salaire (mensuel, annuel, horaire)
experience_required	Expérience requise
category	Catégorie/Métier
sector	Secteur d'activité
application_url	URL pour postuler
source_url	URL source de l'offre
skills	Compétences requises
remote_work	Télétravail possible
is_handicap_accessible	Accessible aux travailleurs handicapés

Mapping entre les sources

Le normalisateur utilise un système de correspondance pour harmoniser les données entre les différentes sources. Voici les principaux mappings mis en œuvre :

Correspondance des champs

Champ normalisé	Adzuna	France Travail
id	id	id
title	title	intitule
description	description	description
company_name	company.display_name	entreprise.nom
location_name	location.display_name	lieuTravail.libelle
latitude	latitude	lieuTravail.latitude
longitude	longitude	lieuTravail.longitude

Champ normalisé	Adzuna	France Travail
date_created	created	dateCreation
date_updated	created (identique)	dateActualisation
contract_type	contract_type	typeContrat
contract_duration	N/A	typeContratLibelle
working_hours	contract_time	dureeTravailLibelleConverti
salary_min	salary_min	Extrait de salaire.libelle
salary_max	salary_max	Extrait de salaire.libelle
category	category.label	romeLibelle
sector	N/A	secteurActiviteLibelle
application_url	redirect_url	contact.urlPostulation
source_url	redirect_url	origineOffre.urlOrigine
skills	N/A	competences[].libelle
is_handicap_accessible	N/A	accessibleTH

Normalisation des valeurs

Types de contrat

Valeur normalisée	Adzuna	France Travail
CDI	permanent	CDI
CDD	contract	CDD
Mission intérimaire	N/A	MIS
Travail saisonnier	N/A	SAI
Profession libérale	N/A	LIB

Horaires de travail

Valeur normalisée	Adzuna	France Travail
Temps plein	full_time	Temps plein
Temps partiel	part_time	Temps partiel

Expérience requise

Valeur normalisée	Adzuna	France Travail
-------------------	--------	----------------

Valeur normalisée	Adzuna	France Travail
Débutant accepté	N/A	D
Expérience souhaitée	N/A	S
Expérience exigée	N/A	E

Périodes de salaire

Valeur normalisée	Adzuna	France Travail
annuel	Par défaut	Déecté via "Annuel" dans salaire.libelle
mensuel	N/A	Déecté via "Mensuel" dans salaire.libelle
horaire	N/A	Déecté via "Horaire" dans salaire.libelle

Idées de l'application

L'objectif n'est pas simplement d'agréger des offres d'emploi, mais de concevoir une application intelligente, personnalisée et proactive qui accompagne réellement les utilisateurs dans leur recherche.

1. Matching intelligent des compétences (Neo4j ?)

- L'utilisateur renseigne ses compétences (ex : Spark, SQL, DataBricks...). → L'application détecte les offres correspondantes, même si les intitulés varient. → Matching basé sur un score personnalisé (nombre de compétences en commun...).

2. Analyse poussées sur les conditions du travail

- Des données essentielles sont présentes dans les descriptions de poste (travail le week-end, port de charges, primes, horaires...). Ce bloc de texte va devoir être analysé en profondeur pour voir ce que nous pouvons en retirer...

Ci-dessous les informations intéressantes qu'on peut retrouver dans chaque description de poste (fait par catégorie) :
 Catégorie d'information Exemples à extraire Utilité dans l'application
 Compétences techniques (Hard Skills) chargement de camions, mise en rayon, plomberie matching métier, scoring profil
 Compétences comportementales (Soft Skills) rigueur, autonomie, esprit d'équipe scoring personnalité, recommandation poste
 Tâches précises / verbes d'action livrer une commande, accueillir le client, poser des éléments sanitaires création d'un référentiel tâches (à rajouter dans la recherche active)
 Conditions physiques station debout, port de charges filtrage contraintes physiques
 Horaires / rythme 6h-13h, travail du lundi au samedi, 39h été filtrage par rythme de travail
 Environnement de travail extérieur, à domicile, sur chantier préférences candidats / compatibilité physique
 Niveau d'expérience 6 mois en GMS vérification pré-requis / scoring profil
 Formation souhaitée formation en plomberie recommandation de formations
 Permis / habilitations permis B obligatoire filtrage par mobilité
 Qualités attendues souriant, dynamique, organisé matching comportemental (pareil que soft skills)
 Avantages / bonus 13e mois, véhicule fourni, prix sur produits score attractivité / tri préférentiel
 Contraintes spécifiques flexibilité, disponibilité week-end
 filtrage contraintes personnelles

3. Recommandation de formations stratégiques

- Identification des compétences manquantes du candidat pour accéder à plus d'opportunités. « si tu avais Spark en plus, tu aurais accès à 45% d'offres supplémentaires » « 70% des entreprises ont Snowflake en compétences demandées »
- Suggestions de formations pertinentes pour booster l'employabilité.

4. Recherche multi-lieux + accessibilité

- L'utilisateur peut rechercher dans plusieurs villes en même temps.
- Prise en compte du trajet domicile-travail (API OpenStreetMap) pour filtrer les offres accessibles en transport ou vélo.

5. Analyse dynamique du marché de l'emploi (API tendance de France Travail)

- Visualisation des tendances : quelles compétences sont en hausse, quels métiers recrutent dans une région.
- Statistiques personnalisées selon le profil du candidat.

6. Alertes intelligentes

- Le système prévient l'utilisateur lorsqu'une offre très pertinente (>80% de matching) apparaît.
- Suivi de l'évolution des offres (dernière mise à jour, pénurie de candidats, etc.).

7. Score d'attractivité d'une offre (Maching Learning)

- Calcul automatique d'un score combinant salaire, distance, compétences requises, avantages...
- Classement des offres selon leur intérêt réel pour l'utilisateur.

Cette vision dépasse la simple recherche d'emploi : on veut créer un copilote intelligent de carrière, capable d'analyser, guider, alerter et recommander de manière proactive.

Particularités des API

API Adzuna

- **Changement important** : Le paramètre `sort_dir` n'est plus supporté par l'API Adzuna depuis sa dernière mise à jour. Le script a été modifié pour utiliser uniquement le paramètre `sort_by` pour le tri.
- **Pagination** : L'API Adzuna utilise un système de pagination basé sur des numéros de page (commençant à 1). Chaque page peut contenir jusqu'à 50 résultats maximum.
- **Limitations** :
 - Maximum de 50 résultats par page
 - Le script gère automatiquement la récupération de plusieurs pages pour atteindre le nombre total de résultats demandés
 - L'accès aux pages au-delà de la limite fixée par Adzuna peut générer des erreurs

API France Travail

- **Pagination** : L'API France Travail utilise un système de pagination par plage (range) au format **p-d** où **p** est l'index (débutant à 0) du premier élément et **d** est l'index du dernier élément.
- **Limitations** :
 - La plage de résultats est limitée à 150 résultats maximum par requête
 - L'index du premier élément ne doit pas dépasser 3000
 - L'index du dernier élément ne doit pas dépasser 3149
 - Le script gère ces limitations en ajustant automatiquement les paramètres de recherche

Installation

Installer python3.11 (python3.8 ne prend pas en charge les bibliothèques)

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt install python3.11 python3.11-venv python3.11-dev

python3.11 --version
```

Créer un environnement virtuel python 3.11

```
python3.11 -m venv .venv
source .venv/bin/activate
```

Installation de Jobmarket

1. Clonez ce dépôt:

```
git clone git@github.com:DataScientest-
Studio/mar25_bootcamp_de_job_market.git
cd mar25_bootcamp_de_job_market
```

2. Installez les dépendances requises:

```
pip install -r requirements
```

3. Configurez les variables d'environnement:

```
cp .env.example .env
```

Modifiez ensuite le fichier **.env** avec vos propres identifiants d'API.

Configuration des API

Adzuna

1. Créez un compte sur [Adzuna for Developers](#)
2. Obtenez vos identifiants d'API (Application ID et API Key)
3. Ajoutez-les dans votre fichier `.env`

France Travail

1. Inscrivez-vous à l'[Emploi Store Développeurs](#)
2. Créez une application et demandez l'accès à l'API Offres d'emploi
3. Obtenez un token d'accès et ajoutez-le dans votre fichier `.env`

Utilisation

Exécution simple

```
python job_data_normalizer.py
```

Variables d'environnement

Vous pouvez configurer les paramètres suivants dans votre fichier `.env`:

- **ADZUNA_APP_ID**: Votre identifiant d'application Adzuna (obligatoire)
- **ADZUNA_APP_KEY**: Votre clé API Adzuna (obligatoire)
- **FRANCE_TRAVAIL_TOKEN**: Votre token d'accès France Travail (obligatoire)
- **DEFAULT_SEARCH_TERMS**: Termes de recherche par défaut (optionnel)
- **DEFAULT_LOCATION_ADZUNA**: Localisation par défaut pour Adzuna (optionnel)
- **DEFAULT_LOCATION_FRANCE_TRAVAIL**: Localisation par défaut pour France Travail (optionnel)
- **DEFAULT_MAX_RESULTS**: Nombre maximum de résultats par source (optionnel)
- **OUTPUT_DIR**: Dossier de sortie pour les fichiers générés (optionnel, par défaut: "job_data")

Résultats

Le script génère trois types de fichiers dans le dossier `data`:

1. `all_jobs_[DATE].csv`: Toutes les offres d'emploi normalisées
2. `adzuna_jobs_[DATE].csv`: Offres d'emploi provenant d'Adzuna
3. `france_travail_jobs_[DATE].csv`: Offres d'emploi provenant de France Travail

Licence

Ce projet est sous licence MIT. Voir le fichier LICENSE pour plus de détails.