



Polytech Montpellier
Parcours Développement Informatique et
Exploitation Opérationnelle (DO)
3^{ème} année

Rapport de Projet : Scout de Stages IA

Serveur MCP & Qualité de Vie des Villes Européennes

Introduction à l'IA

Auteur :
Charlotte LEWIS
charlotte.lewis@etu.umontpellier.fr

Date :
31/01/2026

Université de Montpellier
Place Eugène Bataillon, 34095 Montpellier Cedex 5

Table des matières

Prérequis	3
Résumé	3
1 Architecture système	3
1.1 Pile technologique	3
1.2 Sources de données externes	3
1.3 Flux d'information	4
2 Logique du serveur MCP & gestion des données	4
2.1 Gestionnaire de données Eurostat	4
2.1.1 Méthodologie de classement	4
2.2 Gestionnaire API Tavily	5
2.3 Diagramme d'architecture du serveur MCP	5
3 Implémentation du workflow	6
3.1 Pipeline de traitement	6
4 Configuration et installation	6
4.1 Cloner le dépôt	6
4.2 Configuration de l'environnement	6
4.3 Préparer les données	7
4.3.1 Données Eurostat	7
4.3.2 Données des villes mondiales	7
5 Exécution de l'application	7
5.1 Avec Docker Compose (Recommandé)	7
5.2 Localement pour le développement	8
5.3 Configuration des Identifiants n8n	8
6 Exemples de Résultats	8
7 Défis Techniques & Solutions	9
7.1 Analyse de Données Multidimensionnelles	9
7.2 Gestion de la Charge Utile Discord	10
Glossaire	11
Références	12

Prérequis

⚙️ **Technologies Requisites**

- Python 3.12
- Docker & Docker Compose
- uv (gestionnaire de paquets)


Résumé

Ce projet automatise la recherche d'un stage international en croisant la disponibilité d'offres d'emploi en temps réel avec des indicateurs officiels de qualité de vie (QdV). En établissant un pont technologique entre les opportunités professionnelles et les conditions socio-économiques via le protocole [MCP](#), le système permet aux utilisateurs de trouver des postes dans des villes qui correspondent à leurs valeurs personnelles.

1 Architecture système

L'architecture utilise Docker pour garantir un environnement cohérent entre le serveur MCP Model et le moteur d'automatisation n8n.

1.1 Pile technologique

Composant	Technologie
Runtime	Python 3.12 avec gestionnaire de paquets 
Backend	FastAPI & FastMCP (Interface standardisée avec les LLM)
Orchestration	n8n (Workflow automation et orchestration LLM)
Stockage	SQLite (Données Eurostat) & PostgreSQL (métadonnées n8n)
Conteneurisation	Docker & Docker Compose

1.2 Sources de données externes

Le système s'appuie sur trois sources de données principales :

- [Eurostat](#) : Données statistiques européennes officielles pour les indicateurs au niveau des villes (<https://ec.europa.eu/eurostat/web/cities/database>)
- [API Tavily](#) : Moteur de recherche optimisé pour l'IA, utilisé pour extraire les offres de stages en temps réel
- [SimpleMaps](#) : Base géographique fournissant latitude, longitude et population des villes (<https://simplemaps.com/data/world-cities>)

1.3 Flux d'information

L'architecture suit un modèle [Hub-and-Spoke](#) où n8n agit comme le cerveau central orchestrant trois phases distinctes :

1. **Extraction** : Récupération des données brutes Eurostat et SimpleMaps
2. **Transformation** : Normalisation des scores de qualité de vie via le serveur MCP
3. **Chargement/Action** : Recherche d'offres ciblées et notification finale

***Note sur le MCP** : L'utilisation du Model Context Protocol permet de standardiser la manière dont les outils de recherche sont exposés aux modèles de langage, créant une interface unifiée entre les données structurées et les capacités de raisonnement des LLM.*

2 Logique du serveur MCP & gestion des données

2.1 Gestionnaire de données Eurostat

La classe `EurostatDataManager` est responsable du cycle de vie complet des données des villes, de l'ingestion initiale au classement final. Elle implémente une logique de scoring multicritères.

2.1.1 Méthodologie de classement

Le système calcule un score normalisé pondéré pour chaque ville avec deux couches de logique permettant de gérer le classement.

2.1.1.1 Analyse des couches logiques

Le moteur de décision repose sur une analyse multicritères ([MCA](#)). Pour chaque ville V , le score final S_V est calculé comme suit :

$$S_V = \sum_{i=1}^n w_i \cdot n(I_i)$$

où :

- w_i représente le poids du critère i
- $n(I_i)$ est la valeur normalisée de l'indicateur i

Les trois mécanismes de correction sont :

1. **Correction de polarité** : Pour les indicateurs négatifs (ex : criminalité, pollution, temps de trajet), le système identifie la direction souhaitée et applique une transformation $f(x) = 1 - x$ pour inverser la valeur normalisée.
2. **Normalisation Min-Max** : Les valeurs sont mises à l'échelle selon leurs types d'unités (pourcentages, minutes, devises) pour permettre une comparaison équitable entre des métriques disparates. Chaque indicateur est ramené sur une échelle $[0, 1]$ selon la formule :

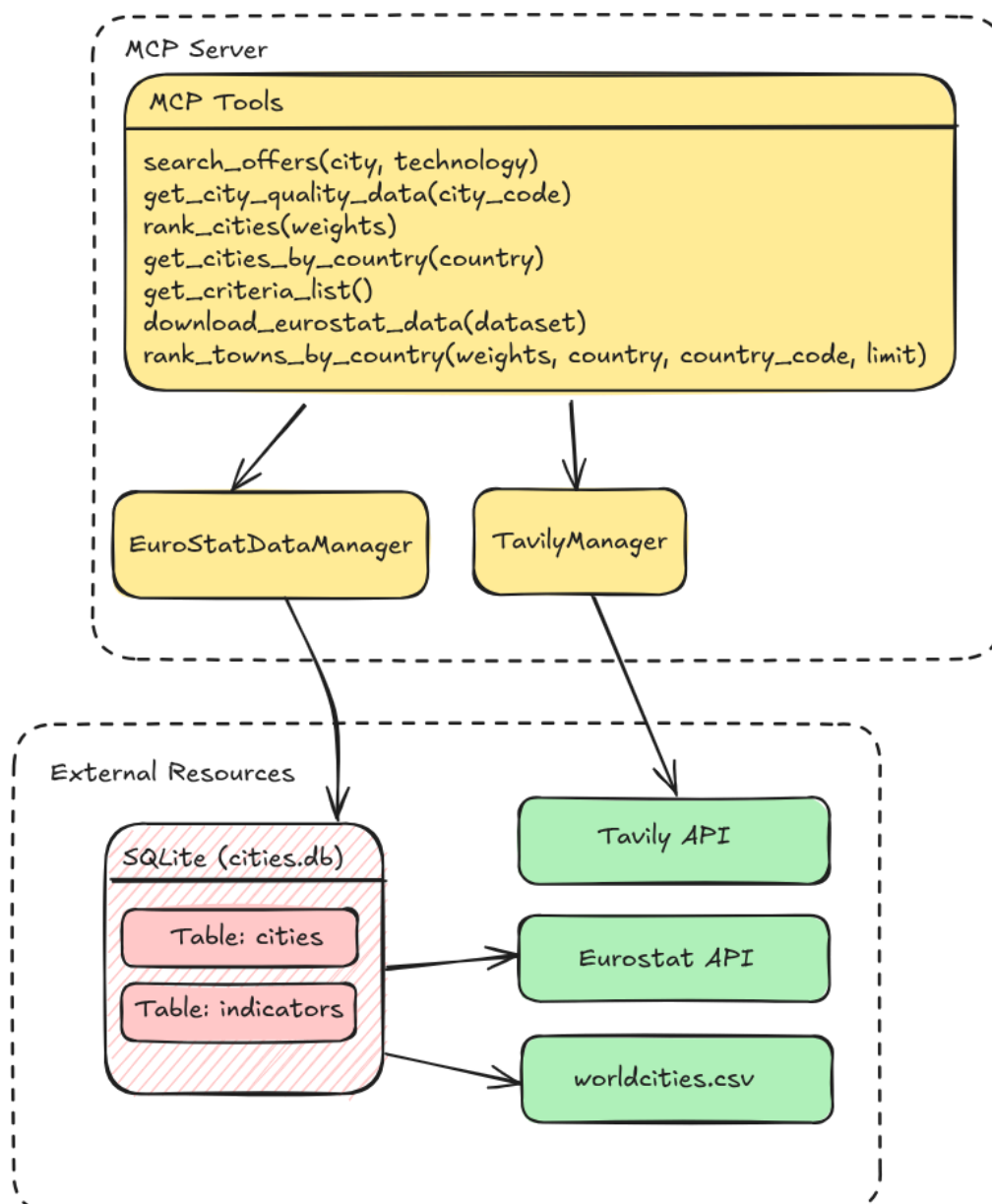
$$n(I_i) = \frac{I_i - I_{\min}}{I_{\max} - I_{\min}}$$

3. **Logique de repli (Fallback)** : Reconnaissant que les données pour les petites municipalités peuvent être rares, le serveur implémente un système de repli récursif. Si aucun indicateur statistique n'est disponible pour une ville demandée, il revient par défaut à un classement basé sur la population pour garantir que l'utilisateur reçoive toujours des suggestions géographiques valides.

2.2 Gestionnaire API Tavily

Le `TavilyManager` sert de pont vers le web en direct. Contrairement aux moteurs de recherche standard, Tavily est optimisé pour le contexte LLM, retournant un contenu propre et hautement pertinent, sans HTML superflu ni bruit informationnel.

2.3 Diagramme d'architecture du serveur MCP



3 Implémentation du workflow

Le workflow n8n suit un pipeline en 8 étapes pour transformer les préférences vagues d'un utilisateur en un rapport structuré et actionnable.

3.1 Pipeline de traitement

1. **Capture des entrées utilisateur** : Collecte les technologies cibles, les pays d'intérêt et les préférences de qualité de vie
2. **Alignement du schéma** : Récupère les critères Eurostat valides via l'outil `get_criteria_list()`
3. **Mapping LLM** : Traduit le langage naturel (ex : « Je déteste le trafic ») en codes compatibles avec la base de données Eurostat
4. **Classement multi-villes** : Identifie les 2 meilleures villes par pays qui correspondent aux critères pondérés
5. **Scraping web ciblé** : Appelle `search_offers()` pour chaque ville identifiée avec des requêtes de recherche optimisées
6. **Extraction d'informations** : Un LLM analyse le HTML/texte brut en un schéma structuré : `Entreprise`, `Poste`, `Compétences`, `Résumé`, `Localisation` et `URL`
7. **Consolidation des données** : Agrège les 10 meilleures offres dans un fichier CSV et un résumé prêt pour Discord
8. **Livraison finale** : Diffuse les résultats via webhook Discord pour une notification instantanée

4 Configuration et installation

4.1 Cloner le dépôt

```
git@github.com:c-r-lewis/n8n-international-internship-scout.git
```

4.2 Configuration de l'environnement

Créez un fichier `.env` à la racine du répertoire du projet. Ce fichier contiendra vos variables d'environnement, y compris les clés API nécessaires.

```
TAVILY_API_KEY="votre_cle_api_tavily_ici"
POSTGRES_PASSWORD="votre_mot_de_passe_postgres_n8n_ici"
```

🔑 Obtention de la Clé API Tavily

Obtenez votre clé API Tavily depuis le site officiel : <https://tavily.com/>

4.3 Préparer les données

L'application s'appuie sur deux sources de données principales qui doivent être configurées avant l'exécution.

4.3.1 Données Eurostat

Le script `eurostat_data_manager.py` est responsable du remplissage de la base de données SQLite `eurostat_data/cities.db`. Pour ingérer les données, vous pouvez utiliser `mcp inspector` et appeler l'outil `download_eurostat_data`.

Exemple de codes de données à ingérer :

- `urb_ctrans` : Transport urbain
- `urb_cecfi` : Culture et infrastructures
- `urb_clivcon` : Conditions de vie
- `urb_cenv` : Environnement
- `urb_clma` : Marché du travail

4.3.2 Données des villes mondiales

Assurez-vous que `ressources/worldcities.csv` est présent dans le projet. Ce fichier fournit les informations géographiques de base (latitude, longitude, population) pour les villes du monde entier.

5 Exécution de l'application

Vous pouvez exécuter l'application en utilisant Docker Compose pour une configuration conteneurisée ou directement via `uvicorn` pour le développement local.

5.1 Avec Docker Compose (Recommandé)

Cette méthode garantit que tous les services s'exécutent dans des conteneurs isolés avec leurs dépendances propres.

```
docker compose up --build
```

L'application devrait alors être accessible à `http://localhost:8000` (ou le port défini dans votre `docker-compose.yml`).

5.2 Localement pour le développement

Assurez-vous d'avoir activé votre environnement virtuel Python (si vous en utilisez un) et installé les dépendances via `uv sync`.

```
uvicorn app:app --reload --host 0.0.0.0 --port 8000
```

Le flag `--reload` active le rechargement automatique lors des modifications du code, ce qui est particulièrement utile pour le développement itératif.

5.3 Configuration des Identifiants n8n

Pour pouvoir exécuter le workflow n8n dans l'interface, vous devrez configurer :

- Un secret **API Mistral Cloud** pour les appels LLM
- Un secret **Webhook Discord** pour la notification finale

🔑 Obtention de la Clé API Mistral

Obtenez votre clé API Mistral depuis le site officiel : <https://console.mistral.ai/home>

6 Exemples de Résultats

Pour une recherche effectuée au **Royaume-Uni** et en **Suisse** ciblant « **l'informatique** » avec un accent sur la **faible criminalité et bons transports**, le système a identifié avec succès des postes clés dans des villes comme Bâle et Southampton.

Les résultats sont livrés sous deux formats :

- **Export Excel** : Fichier structuré avec toutes les offres pertinentes, leurs détails et scores de qualité de vie (<https://github.com/c-r-lewis/n8n-international-internship-scout/blob/main/ressources/spreadsheet.xls>)
- **Notification Discord** : Message formaté automatiquement pour une révision immédiate sur mobile ou desktop

🚀 ****Internship Scout: Top 10 Picks****

Previewing matches across ****13**** different locations:


****1. DRW**** (London, England, United Kingdom)

◆ *Software Developer Intern*

🔗 [View Offer](<https://uk.linkedin.com/jobs/internship-computer-science-jobs-watford>)


****2. Samsara**** (Harrow, England, United Kingdom)

◆ *Software Engineering Intern - London*

 [View Offer](<https://uk.linkedin.com/jobs/internship-computer-science-jobs-watford>)


****3. Infosys**** (London Area, United Kingdom)

◆ *Computer Science Internship Infosys*

 [View Offer](<https://uk.linkedin.com/jobs/internship-computer-science-jobs-watford>)


****4. Hireshire**** (Watford, UK)

◆ *Data Science Intern*

 [View Offer](<https://talents.studysmarter.co.uk/companies/hireshire/watford/data-science-intern-20721937/>)


****5. Dynamic Leads Limited**** (Portsmouth)

◆ *Part Time SDR/ Telemarketing/ Appointment Setter/ Lead Generator/ BDM*

 [View Offer](<https://www.simplyhired.co.uk/search?q=internship&l=gosport%2C+hampshire>)

****6. Roke Manor Research Limited**** (Romsey)

◆ *Summer Internship - National Security*

 [View Offer](<https://www.simplyhired.co.uk/search?q=internship&l=gosport%2C+hampshire>)


****7. Quilter**** (Southampton)

◆ *Early Careers Programme - Technology Intern*

 [View Offer](<https://www.simplyhired.co.uk/search?q=internship&l=gosport%2C+hampshire>)

****8. LiveLink Aerospace**** (Havant)

◆ *Electronics & Embedded Systems Intern*

 [View Offer](<https://www.simplyhired.co.uk/search?q=internship&l=gosport%2C+hampshire>)


****9. ITOL Recruit**** (Gosport, Hampshire)

◆ *Cyber Security Trainee Placement Programme*

 [View Offer](<https://www.cv-library.co.uk/trainee-computer-jobs-in-gosport>)

****10. Boeing**** (Gosport)

◆ *Specialty Engineering Internships*

 [View Offer](<https://jobs.boeing.com/employment/gosport-internship-jobs/185/9287/2635167-6269131-2647554-7290557-2648272/4>)

 ****Full report with all 49 matches attached.****

7 Défis Techniques & Solutions

7.1 Analyse de Données Multidimensionnelles

Problème : Le format [JSON-STAT](#) d'Eurostat est difficile à aplatir en raison de sa structure multidimensionnelle complexe. Les données sont organisées en hypercubes avec des dimensions imbriquées (ville, année, indicateur, unité).

Solution : Développement d'un mappeur personnalisé `_index_to_coords` pour naviguer programmatiquement dans les tailles et IDs de dimensions imbriquées. Cette fonction convertit un index linéaire en coordonnées multidimensionnelles, permettant l'extraction des valeurs.

7.2 Gestion de la Charge Utile Discord

Problème : La limite stricte de 2000 caractères de Discord posait un risque pour le résumé final, particulièrement lorsque plusieurs villes retournaient de nombreuses offres pertinentes.

Solution : Implémentation d'une logique de sélection « round-robin » dans le nœud Code final pour garantir que le résumé incluait un échantillon représentatif de toutes les meilleures villes sans dépasser la limite de caractères. L'algorithme priorise la diversité géographique tout en respectant les contraintes de longueur.

Glossaire

Terme	Définition
MCP	Model Context Protocol - Protocole ouvert permettant aux modèles d'IA d'accéder de manière sécurisée et standardisée à des outils, des données locales et des services tiers.
FastMCP	Framework Python basé sur FastAPI permettant de construire rapidement des serveurs compatibles avec le protocole MCP.
Eurostat	Office statistique de l'Union européenne chargé de fournir des statistiques officielles harmonisées sur l'Europe.
Tavily	Moteur de recherche optimisé spécifiquement pour les agents IA et les LLM, capable de fournir des résultats nettoyés sans HTML superflu.
n8n	Outil d'automatisation de flux de travail (workflow) basé sur les nœuds, permettant d'orchestrer des appels API et de la logique LLM.
MCA	Analyse Multicritères - Méthode d'aide à la décision utilisée pour évaluer et classer des options (villes) en fonction de plusieurs critères contradictoires.
Normalisation Min-Max	Technique statistique ramenant des données de différentes unités sur une échelle commune entre 0 et 1 pour permettre la comparaison.
JSON-STAT	Format de données statistiques multidimensionnelles utilisé par Eurostat pour structurer les données en hypercubes.
Docker Compose	Outil permettant de définir et d'exécuter des applications multi-conteneurs Docker via un fichier de configuration YAML.
SQLite	Système de gestion de base de données relationnelle léger, embarqué directement dans l'application sans serveur séparé.
Modèle Hub-and-Spoke	Architecture où un nœud central (hub) concentre et contrôle les échanges entre plusieurs nœuds périphériques (spokes). Les services ou systèmes périphériques ne communiquent qu'en passant par ce centre, ce qui facilite la centralisation de la sécurité, de la supervision et des règles métiers, au prix d'un point central potentiellement critique.

Références

1. Model Context Protocol Documentation. **Introduction to MCP.** <https://modelcontextprotocol.io>
2. Eurostat Database. **City statistics (urb).** <https://ec.europa.eu/eurostat/web/cities/database>
3. Tavily AI. **Search engine for LLMs.** <https://tavily.com>
4. SimpleMaps. **World Cities Database.** <https://simplemaps.com/data/world-cities>
5. FastAPI Documentation. **FastAPI framework for building APIs.** <https://fastapi.tiangolo.com>
6. n8n Documentation. **Workflow Automation.** <https://docs.n8n.io>
7. Docker Documentation. **Containerization Platform.** <https://docs.docker.com>