# Documentation de l'API de Comparaison de Données (FastAPI)

## 1. Introduction

## 1.1 Objectif de l'API

Bienvenue dans la documentation de l'API de réconciliation de ventes d'Eneo Cameroun. Cette API offre des fonctionnalités pour réconcilier les données de ventes, gérer les utilisateurs, uploader des fichiers vers la base de données, gérer les partenaires, et exécuter des requêtes SQL.

## 1.2 Accès et Authentification

L'accès à l'API nécessite une clé d'API. Veuillez vous inscrire sur notre plateforme pour obtenir votre clé d'API.

## 2. Getting Started

## 2.1 Installation

Aucune installation n'est nécessaire pour utiliser l'API, mais assurez-vous d'inclure votre clé d'API dans les en-têtes des requêtes.

## 2.2 Exemple Rapide

Voici un exemple de création d'un utilisateur :

**```bash**

**curl -X POST "https://api.eneo-cameroon.com/users" -H "Authorization: Bearer YOUR\_API\_KEY" -d '{"username": "john\_doe", "email": "john@example.com"}'**

**```**

## 3. Référence de l'API

## 3.1 Endpoints

## 3.1.1 Utilisateurs

## 3.1.1.1 Création d'un Utilisateur

- **Endpoint** : `POST /users`

- **Description** : Crée un nouvel utilisateur.

- **Paramètres du Corps** :

- **`username` (obligatoire)** : Nom d'utilisateur.

**- `email` (obligatoire)** : Adresse e-mail.

- **Exemple de Réponse :**

**```json**

**{**

**"id": 123,**

**"username": "john\_doe",**

**"email": "john@example.com"**

**}**

**```**

## 3.1.1.2 Mise à Jour d'un Utilisateur

- **Endpoint** : `PUT /users/{user\_id}`

- **Description** :Met à jour les informations d'un utilisateur existant.

- **Paramètres du Corps** :

- `**username**` : Nouveau nom d'utilisateur.

- `**email**` : Nouvelle adresse e-mail.

- **Exemple de Réponse** :

**```json**

**{**

**"id": 123,**

**"username": "john\_doe\_updated",**

**"email": "john\_updated@example.com"**

**}**

**```**

## 3.1.1.3 Suppression d'un Utilisateur

- **Endpoint** :`DELETE /users/{user\_id}`

- **Description** : Supprime un utilisateur.

- **Exemple de Réponse** :

**```json**

**{**

**"message": "Utilisateur supprimé avec succès."**

**}**

**```**

## 3.1.2 Fichiers

## 3.1.2.1 Upload de Fichier

- **Endpoint** : `POST /files/upload`

- **Description** : Upload un fichier vers la base de données.

**- Paramètres du Corps :**

**- `file` (obligatoire)** : Fichier à uploader.

**- Exemple de Réponse :**

**```json**

**{**

**"filename": "example.txt",**

**"message": "Fichier uploadé avec succès."**

**}**

**```**

## 3.1.2.2 Envoi de Fichier vers la Base de Données

- **Endpoint** : `POST /files/to-database`

- **Description** :\*\* Envoie un fichier précédemment uploadé vers la base de données.

**- Paramètres du Corps :**

**- `filename` (obligatoire)** : Nom du fichier à envoyer vers la base de données.

**- Exemple de Réponse :**

**```json**

**{**

**"filename": "example.txt",**

**"message": "Fichier envoyé vers la base de données avec succès."**

**}**

**```**

## 3.1.3 Partenaires

## 3.1.3.1 Création d'un Partenaire

- **Endpoint** : `POST /partners`

- **Description** : Crée un nouveau partenaire.

- **Paramètres du Corps** :

- **`name` (obligatoire)** : Nom du partenaire.

**- Exemple de Réponse :**

**```json**

**{**

**"id": 456,**

**"name": "Partenaire A"**

**}**

**```**

## 3.1.3.2 Suppression d'un Partenaire

- **Endpoint** :`DELETE /partners/{partner\_id}`

- **Description** : Supprime un partenaire.

- **Exemple de Réponse** :

**```json**

**{**

**"message": "Partenaire supprimé avec succès."**

**}**

**```**

## 3.1.4 Requêtes SQL

## 3.1.4.1 Envoi de Requête SQL

- **Endpoint** : `POST /sql`

- **Description** : Envoie une requête SQL vers la base de données.

- **Paramètres du Corps** :

- **`query` (obligatoire)** : Requête SQL à exécuter.

- **Exemple de Réponse** :

**```json**

**{**

**"result": "Succès",**

**"message": "Requête SQL exécutée avec succès."**

**}**

**```**

## 4. Authentification et Autorisation

## 4.1 Types d'Authentification

L'API utilise l'authentification par jeton (Bearer Token). Incluez votre clé d'API dans l'en-tête `Authorization` de chaque requête.

## 5. Exemples d'Utilisation

## 5.1 Scénarios d'Utilisation

Découvrez comment réconcilier les ventes, gérer les utilisateurs, uploader des fichiers, gérer les partenaires, et exécuter des requêtes SQL.

## 6. Gestion des Erreurs

## 6.1 Codes d'Erreur

- 200 OK : La requête a réussi.

- 400 Bad Request : La requête est incorrecte ou manque de paramètres.

- 401 Unauthorized : Clé d'API non valide.

- 404 Not Found : Ressource non trouvée.

- 500 Internal Server Error : Erreur interne du serveur.