

Documentation technique de l'API

Introduction

Cette API REST a été développée en **FastAPI** pour permettre la gestion des objets trouvés, la mise à jour des données de fréquentation des gares, ainsi que l'analyse des conditions météorologiques et des objets trouvés. L'API est sécurisée via un mécanisme d'authentification basé sur des **tokens JWT**, assurant un accès sécurisé aux endpoints.

Accès à l'API :

commandes de lancement API : `<unvicorn main:app --reload>`

étapes pour valider l'authentification afin d'accéder à tous les endpoints de l'API :

1. Aller dans l'endpoint« /token »

> obtenir un token d'accès

- Renseigner le nom d'utilisateur et le mot de passe
 - Si les informations sont correctes, un token sera généré
 - Copier ce token

2. Autorisation via Bearer Token

- Aller dans la section Authorize en haut à droite
 - Dans le champ Bearer, coller le token récupéré précédemment
 - Après cette étape, vous pourrez effectuer des requêtes sur les autres endpoints de l'API

3. Préalable au bon fonctionnement des endpoints en local

- En local, il est nécessaire de se connecter à MSSQL dans VSCode afin de pouvoir interagir correctement avec la base de données. Cette connexion assure le bon fonctionnement des requêtes (SQL) à partir des endpoints de l'API.

Points de Terminaison de l'API

1. « /objets_trouves/ »

>créer un Objet Trouvé

Endpoint: POST /objets_trouves/

Description: Crée un nouvel enregistrement d'objet trouvé.

Request Body: {"date": "YYYY-MM-DD", "type": "string", "gare": "string"}

Réponse: Retourne les détails de l'objet trouvé avec son id.

2. « /count_objets_trouves_gare/ »

>compter les Objets Trouvés

Endpoint: GET /count_objets_trouves_gare/

Description: Compte les objets trouvés dans une période donnée, avec option pour filtrer par gare.

Paramètres:

start_date (obligatoire): Date de début (YYYY-MM-DD)

end_date (obligatoire): Date de fin (YYYY-MM-DD)

gare (optionnel): Nom de la gare

Réponse: Retourne le nombre total d'objets trouvés.

3. « /objets_trouves/ »

>supprimer un objet trouvé

Endpoint: DELETE /objets_trouves/{id}

Description: Supprime un objet trouvé par son identifiant.

Paramètres de Chemin:

id (obligatoire): Identifiant de l'objet

Réponse: Retourne l'identifiant de l'objet supprimé.

4. « /frequentation »

>mettre à Jour la fréquentation d'une gare

Endpoint: PUT /frequentation/{gare}

Description: Met à jour la fréquentation d'une gare pour une année spécifique.

Paramètres de Chemin:

gare (obligatoire): Nom de la gare

year (obligatoire): Année (2021, 2022, ou 2023)

Request Body: {"frequentation": int}

Réponse: Retourne les valeurs mises à jour de la fréquentation.

5. « cloud_sun/ »

>calculer le Poids Pondéré des objets trouvés

Endpoint: POST /cloud_sun/

Description: Calcule le poids pondéré moyen des objets trouvés basé sur les conditions de lumière.

Request Body:

start_date (obligatoire): Date de début (YYYY-MM-DD)

end_date (obligatoire): Date de fin (YYYY-MM-DD)

cloud (obligatoire): Niveau de nuage

sun (obligatoire): Niveau de soleil

Réponse: Retourne le poids pondéré moyen.

