



GloTechT : Application Glossaire Bilingue

Description de la Base de Données

L'application utilise **SQLAlchemy** avec **SQLite** comme backend de base de données. Détails clés de la configuration :

- SQLAlchemy est une bibliothèque puissante du langage de programmation Python qui permet d'interagir avec les bases de données relationnelles de manière intuitive en utilisant un ORM (Object-Relational Mapping) ou des requêtes SQL (Structured Query Language) classiques.
- SQLite est une base de données légère, autonome et open-source, idéale pour des applications nécessitant un stockage intégré et peu complexe.

Modèles de la Base de Données

La base de données contient deux modèles principaux :

1. Modèle de Termes

- Représente les termes du glossaire avec des attributs bilingues (Anglais et Français).
- Les attributs inclus :
 - `tid` : Clé primaire pour identifier chaque terme de manière unique.
 - `domain_en` et `domain_fr` : Domaines pour les termes Anglais et Français (Big Data, Blockchain et Intelligence Artificielle).
 - `subdomains_en` et `subdomains_fr` : Sous-domaines des termes (encore appelés classes).
 - `english_term` et `french_term` : Les termes eux-mêmes.
 - `definition_en` et `definition_fr` : Les définitions de chaque terme en Anglais et en Français.
 - `semantic_label_en` et `semantic_label_fr` : Étiquettes sémantiques.
 - `variant_en` et `variant_fr` : Variantes des termes.
 - `near_synonym_en` et `near_synonym_fr` : Synonymes proches.

- `context_en` et `context_fr` : Contextes d'utilisation du terme.
- `note_en` et `note_fr` : Brève note en rapport avec le terme.
- `syntactic_cooccurrence_en` et `syntactic_cooccurrence_fr` : Les cooccurrences syntaxiques.
- `lexical_relations_en` et `lexical_relations_fr` : Les relations lexicales entre les termes.
- `not_to_be_confused_with_en` et `not_to_be_confused_with_fr` : Les expressions à ne pas confondre avec le terme.
- `frequent_expression_en` et `frequent_expression_fr` : Les expressions fréquentes qui utilisent le terme.
- `phraseology_en` et `phraseology_fr` : La phraséologie du terme.

2. Modèle d'Utilisateur (administrateurs)

- Représente les administrateurs de l'application.
- Attributs inclus :
 - `id` : Clé primaire.
 - `username` : Nom d'utilisateur.
 - `email` : Adresse email de l'utilisateur.
 - `password` : Mot de passe haché.
 - `role` : Rôle de l'utilisateur (par défaut, "user").
- Méthodes incluses :
 - `is_admin()` : Vérifie si l'utilisateur a des privilèges d'administrateur.
 - `to_dict()` : Convertit l'utilisateur en dictionnaire pour la sérialisation JSON (JavaScript Object Notation).

Interactions avec la Base de Données

L'application interagit avec la base de données via plusieurs routes dans `routes.py` :

Interactions liées aux Termes :

- `get_terms()` : Récupérer tous les termes (point de terminaison public).
- `get_term(tid)` : Récupérer un terme spécifique par ID.
- `create_term()` : Créer un nouveau terme (réservé aux administrateurs).
- `update_term(tid)` : Mettre à jour un terme existant (réservé aux administrateurs).
- `delete_term(tid)` : Supprimer un terme (réservé aux administrateurs).
- `search_terms()` : Rechercher des termes dans la base de données.

Interactions liées aux Administrateurs :

- `signup()` : Créer de nouveaux administrateurs.
- `login()` : Authentification des administrateurs.
- `logout()` : Déconnexion des administrateurs.
- `update_password()` : Mettre à jour le mot de passe de l'administrateur.

Authentification et Sécurité

- Utilise `Flask-Login` pour la gestion des sessions utilisateur.

Flask-Login est une extension Flask qui facilite la gestion des sessions utilisateur et l'authentification en fournissant des outils simples pour suivre l'état de connexion.

- Utilise `Flask-Bcrypt` pour le hachage des mots de passe.

Flask-Bcrypt est une extension Flask qui simplifie le hachage sécurisé des mots de passe en utilisant l'algorithme Bcrypt.

- Utilise `Flask-Admin` pour fournir une interface administrateur avec des vues sécurisées pour gérer les Termes et les Utilisateurs, avec des capacités de recherche, de filtrage et de pagination.

Flask-Admin est une extension Flask qui permet de créer rapidement une interface administrateur personnalisable pour gérer les données et les configurations de l'application.

-