

## Documentation : Application de Recherche de Livres avec Chainlit, Azure AI Search et Phi 4 (LM Studio)

### Description de l'application

Cette application permet de rechercher des livres dans une base de données en utilisant Azure AI Search et une interface utilisateur basée sur Chainlit. Elle intègre également un enrichissement des requêtes utilisateur via Phi 4 (LM Studio) pour améliorer la précision des résultats. L'utilisateur peut effectuer des recherches sur plusieurs champs, notamment :

- ISBN (recherche exacte)
- Titre (texte libre)
- Auteur (texte libre)
- Catégorie (recherche exacte)
- Description (texte libre)

Les résultats pertinents sont affichés dynamiquement dans l'interface Chainlit, et l'utilisateur peut demander des détails supplémentaires pour un livre spécifique.

### Fonctionnalités principales

#### 1. Recherche sur plusieurs champs :

- Supporte la recherche par ISBN, Catégorie, Titre, Auteur ou Description.
- Utilise Phi 4 pour reformuler et enrichir les requêtes utilisateur.
- Effectue des recherches fuzzies pour tolérer les fautes de frappe.

#### 2. Affichage dynamique :

- Les informations retournées incluent :
  - ISBN
  - Titre
  - Auteur
  - Description
  - Catégorie
- Possibilité d'obtenir des détails supplémentaires sur un livre spécifique.

#### 3. Suivi de l'état utilisateur :

- Gère l'état utilisateur pour suivre les interactions, comme afficher des détails sur un livre après une recherche initiale.

#### 4. Interface conviviale :

- L'utilisateur saisit une requête et reçoit des résultats formatés avec des instructions claires pour explorer davantage.

## Challenges

### 1. Requêtes utilisateur ambiguës ou mal formulées :

- L'application dépend fortement de Phi 4 pour reformuler les requêtes ambiguës. Une mauvaise reformulation peut entraîner des résultats peu pertinents.

### 2. Tolérance aux fautes :

- Bien que les recherches fuzzies soient prises en charge, des fautes importantes dans les champs clés (comme l'ISBN) peuvent empêcher le système de retourner des résultats pertinents.

### 3. Performance des services Azure :

- La latence des services Azure AI Search peut affecter l'expérience utilisateur, en particulier pour des requêtes complexes ou lorsque l'index contient un grand volume de données.

### 4. Dépendance aux services externes :

- Phi 4 et Azure AI Search sont des services externes. Toute interruption ou configuration incorrecte peut compromettre l'application.

### 5. Gestion des états utilisateur :

- Le suivi de l'état utilisateur sur plusieurs interactions peut devenir complexe, notamment si plusieurs actions simultanées ou non prévues sont exécutées par un utilisateur.

### 6. Qualité des données :

- Les résultats de recherche dépendent directement de la qualité et de l'exactitude des données indexées dans Azure AI Search. Des données incomplètes ou mal formatées peuvent réduire l'efficacité de l'application.

### 7. Scalabilité :

- Lorsque plusieurs utilisateurs accèdent à l'application en même temps, des goulots d'étranglement peuvent survenir, nécessitant une optimisation des ressources et une meilleure gestion des charges.

### 8. Configuration et sécurité :

- Les clés API et les endpoints doivent être protégés pour éviter tout accès non autorisé ou abus des services.

## Structure du projet

### - backend.py :

- Contient la logique principale pour interroger Azure Cognitive Search.
- Intègre Phi 4 pour reformuler les requêtes utilisateur.
- Fournit une interface pour transformer les résultats bruts en un format lisible pour

l'utilisateur.

- app.py :
  - Gère les interactions utilisateur via Chainlit.
  - Implémente la gestion de l'état utilisateur pour des actions telles que la demande de détails supplémentaires.
- chainlit/ :
  - Configuration pour exécuter l'application et intégrer Chainlit.

## Prérequis

1. Services nécessaires :
  - Azure AI Search :
    - Créez un service Azure Cognitive Search via le portail Azure.
    - Configurez un index avec les champs suivants :
      - ISBN, Title, Author, Description, Category.
  - Phi 4 (LM Studio) :
    - Configurez et lancez un serveur Phi 4 (LM Studio) pour traiter les requêtes utilisateur.
2. Environnement Python :
  - Installez Python (version 3.8+ recommandée).
  - Installez les bibliothèques nécessaires avec la commande :

```
pip install azure-search-documents chainlit requests python-dotenv
```
  - Créez et activez un environnement virtuel (recommandé).

## Utilisation

1. Recherche normale :
  - Saisissez une requête texte pour effectuer une recherche libre dans tous les champs.
2. Recherche par ISBN :
  - Saisissez un ISBN exact pour trouver un livre spécifique.
3. Recherche par Catégorie :
  - Saisissez une catégorie pour afficher les livres correspondants.
4. Demande de détails supplémentaires :
  - Après une recherche, tapez "Plus d'informations" et entrez le titre d'un livre pour obtenir des détails.

## Dépannage

- Résultats incorrects ou aucun résultat :
  - Vérifiez la configuration des champs dans l'index Azure Search.
  - Confirmez que Phi 4 est correctement configuré et fonctionne.
  - Assurez-vous que les données sont bien importées dans l'index Azure Search.

- Problèmes de lancement de l'application :
  - Assurez-vous que toutes les bibliothèques nécessaires sont installées.
  - Vérifiez que les fichiers app.py et backend.py sont correctement configurés.
- Problèmes liés à Phi 4 :
  - Vérifiez que le serveur Phi 4 est accessible via l'URL <http://127.0.0.1:1234>.
  - Consultez les logs pour diagnostiquer les erreurs.