Guide d'Ingestion RAG (Retrieval-Augmented Generation)

Ce guide fournit les instructions pour mettre en place et exécuter le script d'ingestion Python, qui prépare votre documentation technique pour une application RAG en utilisant **ChromaDB** comme base de données vectorielle.

1. Prérequis

Assurez-vous d'avoir les éléments suivants installés sur votre système :

- **Python 3.9+**
- **pip** (gestionnaire de paquets Python)

2. Installation des Dépendances

Il est fortement recommandé d'utiliser un environnement virtuel pour isoler les dépendances de ce projet.

Étape 2.1 : Création de l'environnement virtuel

Bash

```
python3 -m venv venv
source venv/bin/activate  # Sous Linux/macOS
# ou
# .\\venv\\Scripts\\activate  # Sous Windows (PowerShell)
```

Étape 2.2 : Installation des paquets

Installez les bibliothèques nécessaires :

- `chromadb` : La base de données vectorielle légère.
- `langchain-community` : Pour les chargeurs de documents (PDF) et l'intégration ChromaDB.
- `pypdf` : Le chargeur de documents PDF.
- `sentence-transformers` : Pour le modèle d'embedding local `all-MiniLM-L6-v2`.

Bash

```
pip install chromadb langchain-community pypdf sentence-transformers
```

3. Préparation des Fichiers

1. **Placez le script** : Assurez-vous que le fichier `ingestion_script.py` se trouve dans un répertoire de projet dédié.
2. **Placez le PDF** : Renommez votre document de spécification (`Documentdespécificationv1-00.pdf`) et placez-le dans le **même répertoire** que le script.

> **Note sur le chemin du fichier :** Le script est configuré pour chercher le fichier nommé `Documentdespécificationv1-00.pdf` dans le répertoire courant. Si vous utilisez un autre nom ou un autre emplacement, vous devrez modifier la ligne `PDF_PATH` dans le script.

4. Exécution du Script d'Ingestion

Le script effectue les opérations suivantes, conformément aux meilleures pratiques RAG :

1. **Nettoyage** : Supprime le répertoire `chroma_db_rag_secops` s'il existe (pour une nouvelle ingestion).
2. **Chargement** : Utilise `PyPDFLoader` pour lire le contenu du PDF.
3. **Chunking** : Utilise `RecursiveCharacterTextSplitter` pour découper le texte en morceaux de **1000 caractères** avec un **chevauchement de 200 caractères**. Ce chevauchement est crucial pour maintenir le contexte lors de la recherche.
4. **Vectorisation** : Utilise le modèle `all-MiniLM-L6-v2` (via `SentenceTransformerEmbeddings`) pour convertir chaque morceau de texte en un vecteur numérique.
5. **Stockage** : Insère les vecteurs et les métadonnées (source, page) dans la collection `secops_documentation` de ChromaDB.

Exécutez le script depuis votre terminal :

Bash

```
python3 ingestion_script.py
```

Résultat Attendu

Après l'exécution, un nouveau répertoire nommé `chroma_db_rag_secops` sera créé dans votre dossier de projet. Ce répertoire contient votre base de données vectorielle prête à être interrogée par votre application RAG (par exemple, un Chatbot SecOps basé sur LangChain ou LlamaIndex).

5. Meilleures Pratiques de Développement Génératif

Le script implémente les meilleures pratiques suivantes :

Pratique	Description	Avantage pour le RAG
Chunking Récursif	Utilisation de `RecursiveCharacterTextSplitter` pour un découpage sémantique plus précis, en essayant différents séparateurs.	Maintient la cohérence des paragraphes et des sections, améliorant la qualité de la récupération.
Chevauchement (Overlap)	Un chevauchement de 200 caractères est défini entre les morceaux.	Assure que le contexte d'une phrase ou d'une idée n'est pas perdu à la frontière d'un *chunk*.
Modèle d'Embedding Local	Utilisation de `all-MiniLM-L6-v2` via `SentenceTransformerEmbeddings` .	Offre un excellent compromis entre vitesse, taille et performance, idéal pour les environnements sans accès constant à des API payantes.
Persistance de la DB	La base de données est stockée localement (`chroma_db_rag_secops`), permettant une réutilisation sans ré-ingestion.	Réduit les coûts et le temps de traitement pour les exécutions futures.

Ce script et ces instructions ont été préparés par Manus AI, en se concentrant sur la robustesse et l'efficacité du pipeline d'ingestion RAG.