



# CaféIA — GraphRAG

## Guide de Déploiement Windows

*LightRAG • Ollama • Mistral API | Workshop UPVD 2026*

⌚ Objectif : déployer un système GraphRAG fonctionnel sur Windows à partir d'une clé USB, sans connexion internet, et démontrer la différence entre RAG classique (naïve) et multi-hop (hybrid).

## Architecture & Stratégie USB

Le déploiement workshop repose sur une séparation claire entre ce qui est fait avant (sur la machine de l'intervenant) et ce qui est fait pendant :

Phase	Où	LLM
<b>Indexation (avant)</b>	Machine intervenant (Linux)	<b>qwen2.5:14b (Ollama local)</b>
<b>Query (pendant workshop)</b>	Laptop étudiant (Windows)	<b>Mistral API (2-5 sec, cloud FR)</b>

💡 Le dossier `rag_storage` (knowledge graph pré-construit) est copié depuis l'USB — les étudiants ne font aucune indexation. Seule la query tourne, via Mistral API.

## PARTIE 1 — Préparer la clé USB (intervenant)

À faire UNE FOIS sur ta machine Linux avant le workshop. Résultat : une clé USB autonome.

### 1.1 — Structure de la clé USB

Créer cette arborescence sur la clé :

```
# Structure clé USB
USB:/
    └── SOURCES/
        ├── python-3.11.9-amd64.exe
        ├── lightrag_hku_api-1.4.9.11-py3-none-any.whl
        └── wheels/           ← toutes les dépendances pip
            └── get-pip.py
```

```

└── rag_storage/      ← knowledge graph PRÉ-CONSTRUIT
    ├── graph_chunk_entity_relation.graphml
    ├── kv_store_*.json
    └── vdb_*.json
└── DOCS/
    ├── clients.csv
    ├── achats_produits.txt
    └── fournisseurs_certifications.txt
├── cafeia_windows.env
└── install_windows.bat
└── start_cafeia.bat

```

## 1.2 — Téléchargements SOURCES

Fichier	URL / Commande	Taille
python-3.11.9-amd64.exe	<a href="https://www.python.org/downloads/releases/python-3119">python.org/downloads/releases/python-3119</a>	~25 MB
lightrag-hku[api]	pip download lightrag-hku[api] -d wheels/	~150 MB
rag_storage/	Copier depuis ta machine Linux	~50 MB
clients.csv	Fichier fourni (dans ce guide)	< 1 KB
achats_produits.txt	Fichier fourni (dans ce guide)	< 10 KB
fournisseurs_certifications.txt	Fichier fourni (dans ce guide)	< 10 KB
cafeia_windows.env	Fichier fourni (dans ce guide)	< 1 KB
install_windows.bat	Fichier fourni (dans ce guide)	< 1 KB
start_cafeia.bat	Fichier fourni (dans ce guide)	< 1 KB

Commande pip download (à lancer sur Linux avant le workshop) :

```

pip download "lightrag-hku[api]" -d /media/usb/SOURCES/wheels/
# Crée tous les .whl nécessaires pour install offline

```

## PARTIE 2 — Fichiers de configuration

### 2.1 — cafeia\_windows.env

Configuration clé : LLM\_BINDING=openai pointant sur Mistral API. Pas d'Ollama nécessaire côté étudiant. Le rag\_storage est pré-construit — aucune indexation ne se fait.

```
# cafeia_windows.env – copier sur la clé USB
HOST=0.0.0.0
PORT=9621
WORKERS=1
TIMEOUT=120
LOG_LEVEL=INFO

# LLM : Mistral API (OpenAI-compatible) – RAPIDE, no GPU
LLM_BINDING=openai
LLM_BINDING_HOST=https://api.mistral.ai/v1
LLM_BINDING_API_KEY=METTRE_CLE_MISTRAL_ICI
LLM_MODEL=mistral-small-latest
LLM_TIMEOUT=60
MAX_ASYNC=4
MAX_TOKENS=8192

# Embedding : mxbai-embed-large via Ollama (optionnel – cache déjà OK)
# Si rag_storage complet = embeddings déjà calculés = pas besoin
EMBEDDING_BINDING=ollama
EMBEDDING_BINDING_HOST=http://localhost:11434
EMBEDDING_MODEL=mxbai-embed-large
EMBEDDING_DIM=1024

# Query params
HISTORY_TURNS=5
COSINE_THRESHOLD=0.2
TOP_K=40
MAX_GRAPH_NODES=500
CORS_ORIGINS=*
```

⚠ Remplacer METTRE\_CLE\_MISTRAL\_ICI par la vraie clé avant de copier sur les USB. La clé Mistral est valide pour tous les étudiants — prévoir le quota.

## 2.2 — install\_windows.bat

```
# install_windows.bat – à lancer EN ADMIN une seule fois
@echo off
echo === Installation CaféIA LightRAG ===

# 1. Python (si pas installé)
SOURCES\python-3.11.9-amd64.exe /quiet InstallAllUsers=1 PrependPath=1

# 2. Créer dossier de travail
mkdir C:\cafeia
cd C:\cafeia

# 3. Venv
python -m venv venv
```

```
call venv\Scripts\activate.bat

# 4. Install lightrag offline depuis wheels USB
pip install --no-index --find-links=%~dp0SOURCES\wheels lightrag-hku[api]

# 5. Copier rag_storage et config
xcopy /E /I %~dp0rag_storage C:\cafeia\rag_storage
copy %~dp0cafeia_windows.env C:\cafeia\.env

echo === Installation terminée ===
pause
```

## 2.3 — start\_cafeia.bat

```
# start_cafeia.bat – lancer à chaque démarrage
@echo off
cd C:\cafeia
call venv\Scripts\activate.bat
echo Demarrage CaféIA GraphRAG...
echo WebUI disponible sur : http://localhost:9621
start http://localhost:9621
lightrag-server
pause
```

# PARTIE 3 — Installation sur laptop étudiant (Windows)

Prérequis minimum : Windows 10/11, 8 GB RAM, 5 GB disque libre. GPU non requis.

## Étapes séquentielles

1. **ÉTAPE 1** — Brancher la clé USB
2. **ÉTAPE 2** — Ouvrir une invite de commande EN ADMINISTRATEUR  
Touche Windows → taper cmd → clic droit → Exécuter en tant qu'administrateur
3. **ÉTAPE 3** — Naviguer vers la clé USB et lancer l'installation

```
E:\install_windows.bat
# Remplacer E: par la lettre de la clé USB
```

4. **ÉTAPE 4** — Attendre la fin de l'installation (~2 minutes)  
Le script installe Python, le venv, lightrag, et copie le rag\_storage.
5. **ÉTAPE 5** — Lancer CaféIA
6. **ÉTAPE 6** — Le navigateur s'ouvre automatiquement sur http://localhost:9621

 Si Python est déjà installé sur le laptop, l'étape 1 du bat sera ignorée automatiquement. Aucun risque de conflit.

## PARTIE 4 — Utilisation pendant le Workshop

---

### 4.1 — Interface LightRAG WebUI

Onglet 'Documents' : vérifier que les 3 documents sont listés (déjà indexés — status DONE).

Onglet 'Knowledge Graph' : visualiser le graphe d'entités (fournisseurs, clients, produits).

Onglet 'Retrieval' : interface de requête avec sélection du mode.

### 4.2 — Questions de démo multi-hop

Ces questions sont conçues pour échouer en mode naïve et réussir en mode hybrid. C'est le cœur de la démonstration GraphRAG.

#### Question 1 — 3 hops (client → achat → fournisseur → certification)

Quels clients ont acheté des produits d'un fournisseur certifié AB ?

Mode naïve : retourne des chunks sur les certifications OU les achats, jamais croisés.

Mode hybrid : traverse client → achat\_produit → fournisseur → certification AB.

#### Question 2 — 2 hops (client professionnel + fournisseur spécialisé)

Quels clients professionnels ont commandé des systèmes d'irrigation ?

#### Question 3 — comparaison attributs fournisseurs

Quel fournisseur certifié Bio livre en moins de 72h ?

#### Question 4 — 4 hops (le plus difficile)

Qui a acheté des semences d'un fournisseur certifié Demeter en Occitanie ?

#### Question simple — validation de base

Quel est le numéro de téléphone de Marie Dupont ?

Doit répondre : 06.12.34.56.78 avec référence clients.csv

### 4.3 — Utiliser les modes via prefix

Dans le champ de requête, préfixer le message pour forcer le mode :

```
/naive Quels clients ont acheté Bio ?  
/hybrid Quels clients ont acheté Bio ?  
/local Quels clients ont acheté Bio ?
```

```
/global Quels clients ont acheté Bio ?
```

## PARTIE 5 — Troubleshooting Windows

Erreur : 'lightrag-server' n'est pas reconnu

```
C:\cafeia\venv\Scripts\lightrag-server.exe  
# Utiliser le chemin complet si PATH non configuré
```

Erreur : port 9621 déjà utilisé

```
netstat -ano | findstr :9621  
taskkill /PID <pid> /F
```

Réponse 'créé par Anthropic' au lieu de qwen

LightRAG détecte ANTHROPIC\_API\_KEY dans l'environnement. Solution :

```
set ANTHROPIC_API_KEY=  
set OPENAI_API_KEY=  
lightrag-server
```

Documents non trouvés dans l'interface

Vérifier que le dossier rag\_storage est bien dans C:\cafeia\rag\_storage

Vérifier que .env contient les bons chemins (pas de RAG\_DIR custom).

Ollama non installé — erreur embedding

Si rag\_storage est complet (tous les .json présents), aucune re-indexation n'est nécessaire et Ollama n'est pas requis. L'embedding ne se lance qu'à l'indexation.

## Résumé — Checklist intervenant

7.  Indexer les 3 documents sur ta machine Linux (qwen2.5:14b)
8.  Copier rag\_storage/ sur la clé USB
9.  Télécharger les wheels offline : pip download lightrag-hku[api] -d wheels/
10.  Mettre la clé Mistral dans cafeia\_windows.env
11.  Tester install\_windows.bat sur un Windows test
12.  Tester les 4 questions multi-hop — vérifier hybrid vs naive
13.  Préparer 1 clé USB par poste étudiant (ou partage réseau local)

 Total USB : ~250 MB. Une clé 1 GB suffit. Tester sur Windows propre avant le workshop.

Mensaflow × UPVD — 2026 — *confidentiel usage pédagogique*