Brugervejledning til Ingest Script

Script: ingest_batch_v3.py
Version: 3.0
Formål: Automatisk indlæsning af dokumenter til RAG-databasen

---

1. Oversigt

Dette script håndterer automatisk synkronisering af dokumenter mellem filsystemet og ChromaDB vektordatabasen. Scriptet sikrer at:

- Nye dokumenter bliver embeddet og tilføjet til databasen
- Ændrede dokumenter bliver opdateret (gamle data slettes først)
- Slettede dokumenter fjernes fra databasen
- Ingen dubletter opstår

---

2. Mappestruktur

Før du kører scriptet, skal du have følgende mappestruktur:

```
projekt/
|
|-- prompts/                    <- System prompts (trin 1)
|   |-- 1_it_servicedesk.txt
|   |-- 2_hr_support.txt
|   |-- 3_salg_info.txt
|
|-- documents/                  <- Dokumenter (trin 2)
|   |-- it_servicedesk/
|   |   |-- manual.pdf
|   |   |-- faq.docx
|   |
|   |-- hr_support/
|   |   |-- personalepolitik.pdf
|   |
|   |-- salg_info/
|       |-- produktkatalog.pdf
|
|-- chroma_collections/         <- Database (oprettes automatisk)
|
```

```
|-- ingest_batch_v3.py          <- Dette script
|-- .env                        <- OpenAI API key
```

---

3. Trin-for-trin Vejledning

Trin 1: Opret System Prompts

Læg dine systemprompt-filer i mappen `prompts/`

Filnavns-format: `{nummer}_{collection_navn}.txt`

Eksempler: - `1_it_servicedesk.txt` - `2_hr_support.txt` - `3_kundeservice.txt`

Scriptet opretter automatisk tilsvarende mapper i `documents/`: - `documents/it_servicedesk/` - `documents/hr_support/` - `documents/kundeservice/`

Trin 2: Tilføj Dokumenter

Læg dine dokumenter i den relevante mappe under `documents/`

Understøttede formater: | Format | Filtype | |———–|———| | PDF | .pdf | | Word | .docx, .doc | | Markdown | .md | | Tekst | .txt |

Eksempel:

```
documents/it_servicedesk/
|-- bruger_manual.pdf
|-- FAQ.docx
|-- hurtig_guide.md
```

Trin 3: Kør Scriptet

```
python ingest_batch_v3.py
```

Scriptet vil:
1. Synkronisere mapper med prompt-filer
2. Skanne alle dokumenter
3. Opdatere databasen

---

4. Automatisk Håndtering

4.1 Nye Dokumenter

Når et NYT dokument lægges i en mappe:

```
[Før]                          [Efter]
documents/it/                   documents/it/
|-- manual.pdf                  |-- manual.pdf
                                |-- ny_guide.pdf  <- NY FIL
```

Scriptet gør:
1. Registrerer ny fil (ny_guide.pdf)
2. Beregner SHA256 hash
3. Splitter i chunks (500 tegn)
4. Genererer embeddings via OpenAI
5. Gemmer i ChromaDB med metadata

Output:

```
📄 ny_guide.pdf
   📕 PDF loaded (12 sider)
   📊 Chunks: 45
   ✅ [INGEST] Gemt
```

4.2 Ændrede Dokumenter

Når et EKSISTERENDE dokument ændres:

```
[Før]                          [Efter]
manual.pdf (hash: abc123)       manual.pdf (hash: xyz789)  <- ændret
```

Scriptet gør:
1. Sammenligner SHA256 hash med gemt hash
2. Opdager at filen er ændret
3. Sletter alle gamle chunks fra databasen
4. Genindlæser dokumentet med nye data

Output:

```
📄 manual.pdf
   🔄 [UPDATE] Fil aendret - genindlæser...
   📕 PDF loaded (15 sider)
   📊 Chunks: 52
   ✅ [UPDATE] Gemt
```

VIGTIGT: Gamle data slettes Først - ingen dubletter!

4.3 Slettede Dokumenter

Når et dokument SLETTES fra mappen:

```
[Før]                          [Efter]
documents/it/                   documents/it/
```

```
|-- manual.pdf                  |-- manual.pdf
|-- gammel_guide.pdf            (slettet)
```

Scriptet gør:
1. Sammenligner filer i mappen med dokumenter i databasen
2. Finder "gammel_guide.pdf" i DB men ikke i mappe
3. Sletter alle chunks for dette dokument fra databasen

Output:

```
🧹 Oprydning: 1 forældede dokumenter fjernet
🗑️  SLETTET fra database: gammel_guide.pdf
```

4.4 Uændrede Dokumenter

Når et dokument er uændret:

Scriptet gør:
1. Sammenligner SHA256 hash
2. Hash matcher - springer over

Output:

```
📄 manual.pdf
   ✅ [SKIP] Uaendret
```

---

5. Metadata der Gemmes

For hver chunk gemmes følgende metadata:

Felt	Beskrivelse	Eksempel
document	Filnavn	"manual.pdf"
document_id	Unik ID	"manual_pdf"
collection	Collection navn	"it_servicedesk"
sha256	Fil hash	"abc123..."
page	Sidenummer	5
chunk_index	Chunk nummer (0-baseret)	12
total_chunks	Antal chunks i alt	45
position	Position i dokument	"start/middle/end"
keywords	Udtrukne noegleord	"server,netvaerk,login"
ingested_at	Tidspunkt for ingest	"2026-01-31T10:00:00"

Felt	Beskrivelse	Eksempel
file_modified_at	Fil sidst aendret	“2026-01-30T15:30:00”
sourceUrl	Link til dokument	“https://server.dk/files/manual.pdf”

---

6. Konfiguration

I scriptet kan du tilpasse:

```
# Mapper
DOCS_BASE_DIR = "documents"            # Dokument-mappe
CHROMA_BASE_DIR = "chroma_collections" # Database-mappe
PROMPT_DIR = "prompts"                 # Prompt-mappe

# Chunking
CHUNK_SIZE = 500                       # Tegn per chunk
CHUNK_OVERLAP = 50                     # Overlap mellem chunks

# URL til dokumenter (til sourceUrl metadata)
SERVER_BASE_URL = "https://server.dk/files/"

# Ekskluder mapper fra processing
EXCLUDED_COLLECTIONS = [
    "retsinfo",    # Disse mapper ignoreres
]
```

---

7. Eksempel på Kørsel

```
============================================================
🔄 SYNKRONISERER MAPPER MED PROMPT-FILER
============================================================
   ✅ EXISTS:   it_servicedesk/ (fra 1_it_servicedesk.txt)
   📁 OPRETTET: hr_support/ (fra 2_hr_support.txt)

📋 2 collections synkroniseret fra prompts
============================================================

============================================================
📥 STARTER DOKUMENT INGEST
============================================================

╔══════════════════════════════════════════════════════════
```

```
╔═══════════════════════════════════════════════════════════╗
║ 📚 Collection: it_servicedesk                             ║
╚═══════════════════════════════════════════════════════════╝
  📋 Filer fundet: 3

  🧹 Oprydning: 1 foraeldede dokumenter fjernet

  📄 manual.pdf
    ✅ [SKIP] Uaendret
  📄 faq.docx
    🔄 [UPDATE] Fil aendret - genindlaeser...
    📘 Word dokument loaded
    📊 Chunks: 28
    ✅ [UPDATE] Gemt
  📄 ny_guide.pdf
    📕 PDF loaded (5 sider)
    📊 Chunks: 18
    ✅ [INGEST] Gemt

═══════════════════════════════════════════════════════════
✅ INGEST FAERDIG
═══════════════════════════════════════════════════════════

📊 Statistik:
   • Collections fundet:       2
   • Dokumenter processeret:   2
   • Dokumenter sprunget over: 1
   • Dokumenter slettet fra DB: 1
   • Total chunks oprettet:    46
```

---

8. Fejlfinding

Problem: "Kunne ikke importere document loaders"

Løsning:

```
pip install langchain-community pypdf docx2txt unstructured markdown
```

Problem: "OPENAI_API_KEY ikke sat"

Løsning: Opret `.env` fil med:

```
OPENAI_API_KEY=sk-proj-din-api-noegle-her
```

Problem: Dokumenter bliver ikke opdateret

Årsag: Filen er uændret (samme SHA256 hash)

Løsning: Scriptet springer bevidst uændrede filer over for at spare API-kald.

---

9. Bekræftelse af Funktionalitet

Baseret på kodeanalyse bekræftes følgende:

Funktion	Status	Implementering
Nye dokumenter indlaeses	✓	`ingest_document()` linje 353-439
AEndrede dokumenter opdateres	✓	SHA256 check + slet + geningest, linje 372-378
Gamle data slettes først	✓	`vectordb.delete()` linje 378
Slettede filer fjernes fra DB	✓	`cleanup_deleted_documents()` linje 299-350
Ingen dubletter	✓	Hash-check forhindrer dubletter
Prompt-synkronisering	✓	`sync_folders_with_prompts()` linje 65-114

Dokument slut

Genereret: Januar 2026