

생성형 AI 기반 RAG 개발자 과정

문서 파싱 및 임베딩 (색인)

Session 19 / 33 — Day 3

저장된 파일을 검색 가능한 벡터로 바꾼다

박수현 · (사)한국정보공학기술사회 · 2026

LLM 이론 · LangChain · ChromaDB CRUD · Agent — 4일 핸드온

이번 시간을 마치면

색인 파이프라인

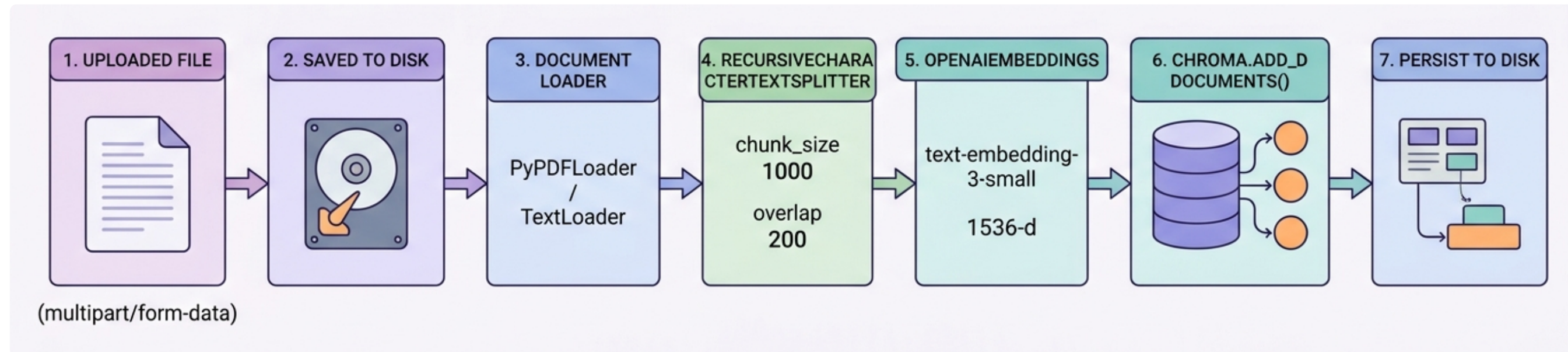
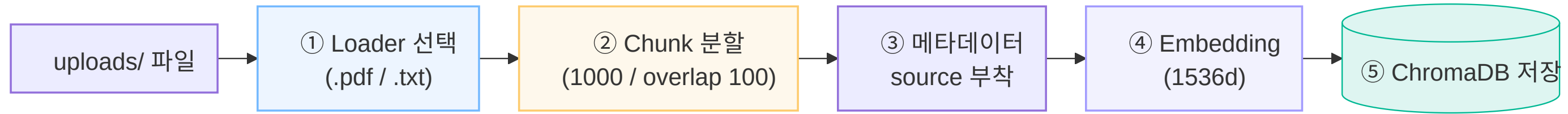
- 확장자별 **Loader** 자동 선택
- **Chunk** 생성 (splitter)
- **Embedding** → ChromaDB 저장

서비스로 캡슐화

- `add_document(path)` 한 함수로
- 메타데이터(source) 부착
- route 에서 호출만

 목표: 파일 경로 입력 → 검색 가능 상태를 단일 서비스 함수로 캡슐화한다.

파일 → 검색 가능한 벡터



🖼 인덱싱 · 업로드(multipart) → 저장 → Loader → Splitter(1000/100) → OpenAIEmbeddings(1536d) → Chroma.add_documents → 영속화

💡 이 5단계가 `vectorstore.add_document()` 의 내부 구현이다. route 는 이를 **한 줄로 호출**한다.

확장자에 맞는 로더 (스니펫)

```
from langchain_community.document_loaders import (
    TextLoader, PyPDFLoader, Docx2txtLoader)

def load_file(path):
    ext = os.path.splitext(path)[1].lower()
    if ext == ".pdf": return PyPDFLoader(path).load()
    if ext == ".docx": return Docx2txtLoader(path).load()
    return TextLoader(path, encoding="utf-8").load()
```

- 기본은 **PDF/TXT** — 확장자 분기로 **형식별 로더** 자동 적용
- 출력은 모두 **Document** 리스트 → 이후 단계는 형식과 무관
- **.docx** 분기는 **확장 예시** — **Docx2txtLoader** 추가로 형식 확장

📁 참고: [8.web_app/.../services/vectorstore.py](#)

add_document 단일 함수 (스니펫)

```
splitter = RecursiveCharacterTextSplitter(
    chunk_size=1000, chunk_overlap=100)    # 1000자 / 100 겹침

def add_document(path):
    docs = load_file(path)                # ① 로드
    chunks = splitter.split_documents(docs) # ② 분할
    for c in chunks:                       # ③ 메타데이터
        c.metadata["source"] = os.path.basename(path)
    store.add_documents(chunks)             # ④⑤ 임베딩+저장
    return {"source": path, "chunks": len(chunks)}
```

- `store` = 영속 ChromaDB (OpenAIEmbeddings·1536차원으로 자동 임베딩+저장)
- `chunk_overlap=100` = 경계에 걸친 문장 누락을 줄이는 겹침
- `source` 메타데이터 = **목록·삭제·출처** 추적의 키

🔑 route 는 `add_document(saved_path)` 한 줄만 호출한다.

두 설계의 트레이드오프

A 통합 (POST /files)

- 업로드와 동시에 색인 수행
- **단순**, 한 번의 요청으로 완료
- 대용량은 응답 지연 가능

B 분리 (/files → /api/index)

- 업로드와 색인을 **분리**
- 진행률 표시·재색인에 유연
- 구현 복잡도 증가

💡 학습·소규모 환경에서는 **A 통합**으로 충분하다. 본 코스 데모도 업로드 시 즉시 색인한다.

핵심 5가지

1.  색인 = **Loader** → **Chunk** → **메타데이터** → **Embedding** → **저장**
2.  확장자로 **Loader** 자동 선택 (PDF/TXT 기본 · DOCX 는 분기 한 줄 추가)
3.  `add_documents` 가 **임베딩+저장**을 한 번에
4.  **source** 메타데이터 = 목록·삭제·**출처** 추적의 키
5.  업로드와 색인은 **통합 또는 분리** — 규모에 따라 선택