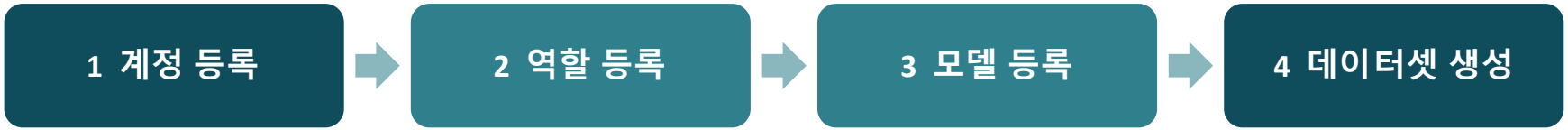
# Entology 교육용 사용 매뉴얼

강사 준비사항과 훈련생 실습 가이드 | 초안 2026-11

# 전체 흐름

## Part 1. 강사 준비사항 (교육 전, 관리자 계정)




## Part 2. 훈련생 실습 (배부받은 계정)




**참고** Part 1은 강사가 한 번만 준비합니다. 훈련생은 Part 2만 진행합니다.

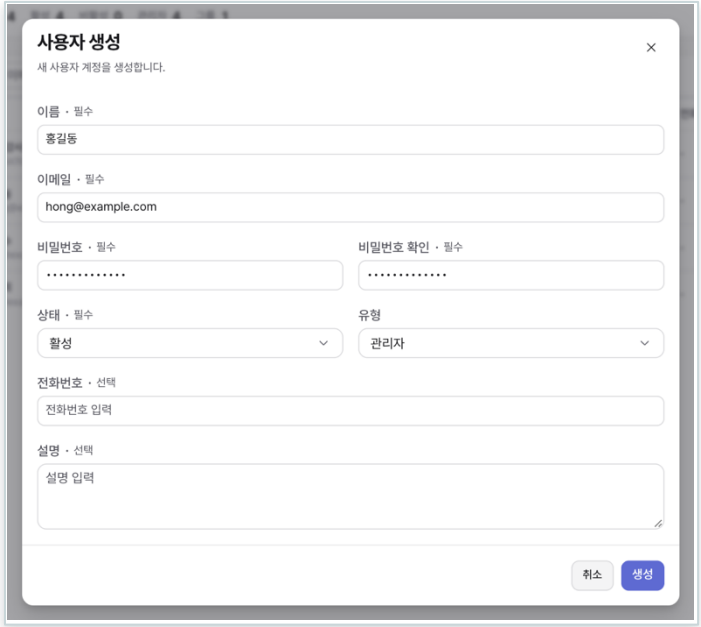
# Part 1. 강사 준비사항

교육 전에 강사가 관리자 계정으로 한 번 준비합니다.
계정, 역할, AI 모델, 검색용 데이터셋을 만듭니다.

# 1. 훈련생 계정 등록

Administration > 계정 및 그룹

1. **그룹 추가**
   [그룹 추가] → '그룹 생성' 창에 그룹 이름 입력 (예: 2026-11 교육반)
2. **사용자 추가**
   [사용자 추가] → '사용자 생성' 창에 이름, 이메일, 비밀번호(8자 이상)
3. **상태와 유형 지정 후 [생성]**
   상태: 활성 / 유형: 관리자
4. **인원수만큼 반복**
   이메일과 초기 비밀번호를 정리해 교육 당일 배부




**참고** 유형이 메뉴 접근 범위를 정합니다. 훈련생이 워크플로를 만들고 역할을 할당하려면 '관리자'로 등록합니다.

# 2. 역할 등록

Administration > 역할 및 권한

**1** **역할 생성**

[역할 생성] → 역할 이름 (예: 교육생), 설명 → [생성]

**2** **역할 상세 열기**

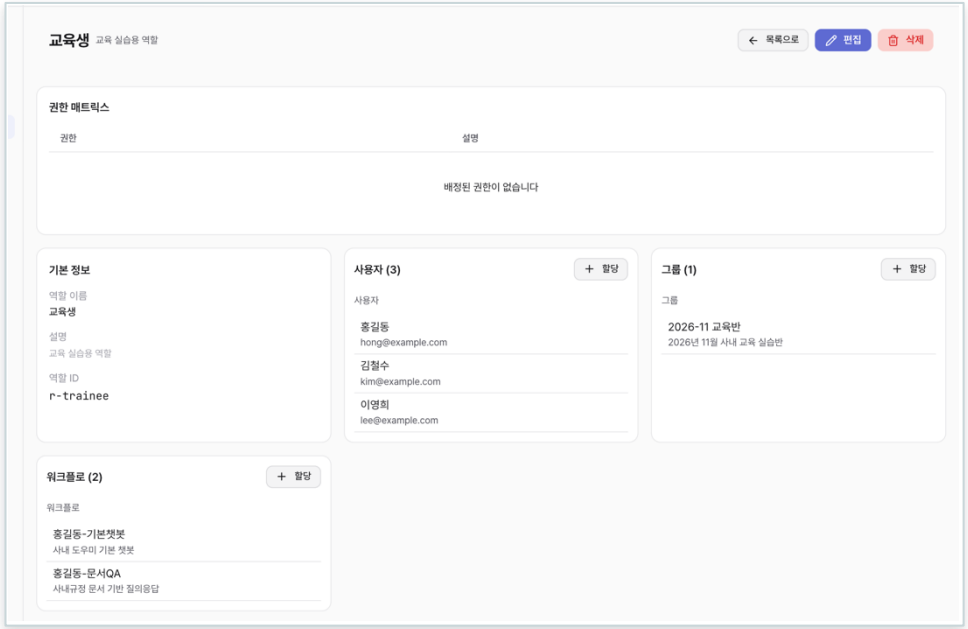
목록에서 방금 만든 역할을 선택

**3** **사용자 할당**

사용자 카드의 [할당] → 훈련생 전원 선택

**4** **그룹 할당**

그룹 카드의 [할당] → 1단계에서 만든 그룹 선택



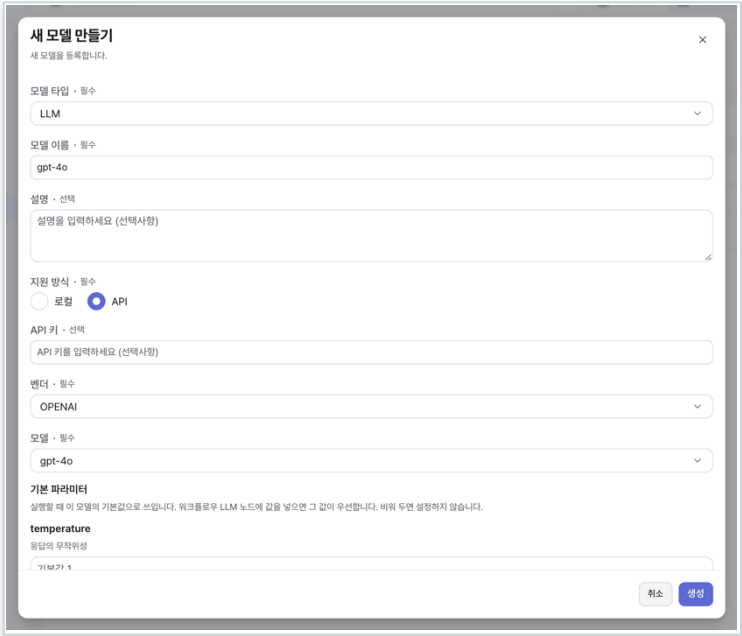

**참고** 역할은 '어떤 앱(워크플로)을 쓸 수 있는가'를 정합니다. 워크플로 연결은 훈련생이 Part 2의 6단계에서 합니다.

# 3-1. LLM과 임베딩 모델 등록

Administration > AI 모델 > 모델 카탈로그

[모델 추가] → '새 모델 만들기' 창에서 각각 등록

| 항목 | LLM | 임베딩 |
|---|---|---|
| **모델 타입** | LLM | 임베딩 |
| **모델 이름** | gpt-4o | text-embedding-3-large |
| **지원 방식** | API | API |
| **벤더** | OPENAI | OPENAI |
| **모델** | gpt-4o | text-embedding-3-large |
| **API 키** | OpenAI API 키 | OpenAI API 키 |




**참고** 모델 화면에는 연결 테스트가 없습니다. API 키는 4-2 벡터화 실행으로 확인합니다.

# 3-2. 기본 모델 지정

Administration > AI 모델 > 기본 모델

**1** **[기본 모델] 탭 열기**

AI 모델 화면의 두 번째 탭 (첫 번째는 모델 카탈로그)

**2** **Embedding 지정**

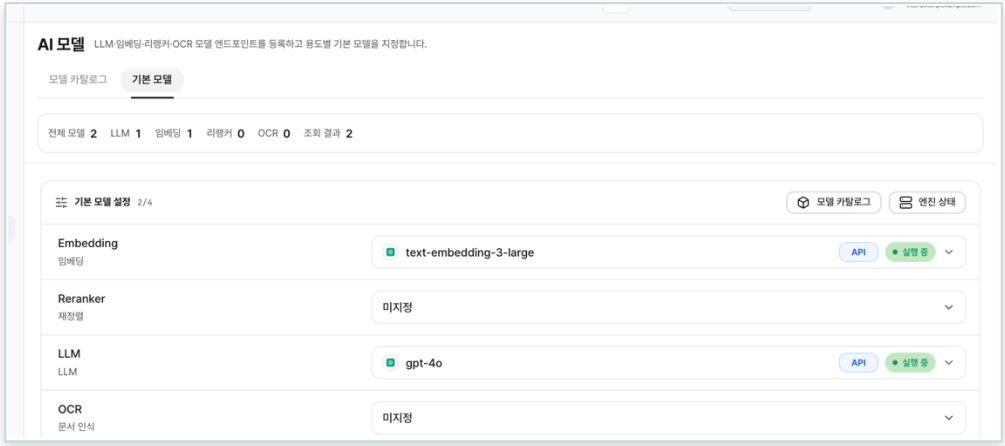
3-1에서 등록한 text-embedding-3-large 선택

**3** **LLM 지정**

3-1에서 등록한 gpt-4o 선택

**4** **나머지는 비워 둠**

Reranker, OCR은 이번 교육에서 쓰지 않음

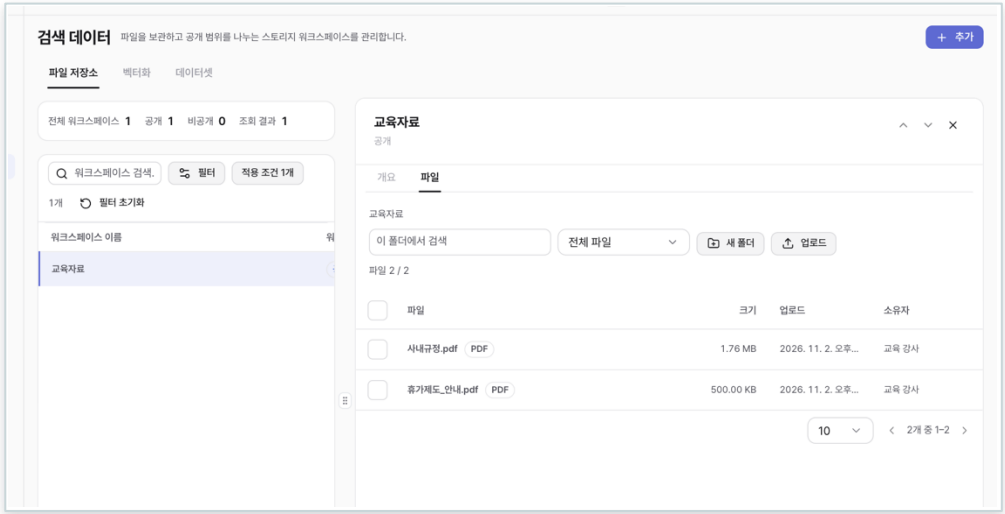



**참고** 기본 모델은 워크플로 노드와 벡터화에 자동으로 들어가지 않습니다. 각 화면에서 모델을 직접 고릅니다.

# 4-1. 워크스페이스 생성과 파일 업로드

Data Fabric > 검색 데이터 > 파일 저장소

1. **워크스페이스 추가**
   [추가] → 워크스페이스 이름, 유형: 공개
2. **워크스페이스 열기**
   목록에서 방금 만든 워크스페이스 선택
3. **파일 업로드**
   상세의 [파일] 탭 → [업로드] → 교육용 문서 선택
4. **업로드 확인**
   목록에 파일이 표시되는지 확인




**참고** 화면에서는 파일 저장소를 '워크스페이스'라고 부릅니다. 훈련생도 쓸 수 있도록 '공개'로 만듭니다.

# 4-2. 벡터화 실행

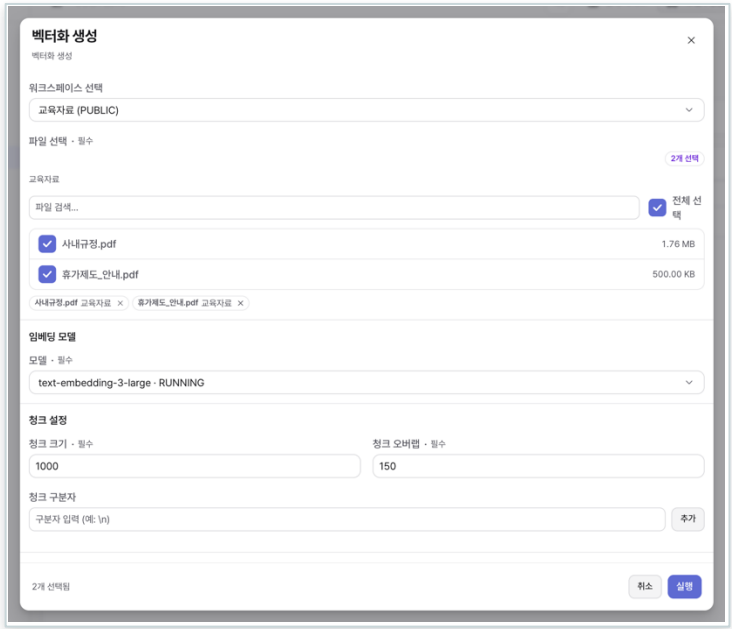
Data Fabric > 검색 데이터 > 벡터화

1. **벡터화 생성 창 열기**
   벡터화 탭에서 [추가]

2. **대상 선택**
   4-1의 워크스페이스와 업로드한 파일 선택

3. **설정**
   임베딩 모델: text-embedding-3-large / 청크 크기 1000, 청크 오버랩 150 (기본값)

4. **[실행] 후 완료 대기**
   '벡터화가 완료되었습니다' 알림을 확인




**참고** 벡터화는 백그라운드에서 진행됩니다. 완료되기 전에는 데이터셋 생성 목록에 파일이 나오지 않습니다.

# 4-3. 데이터셋 생성

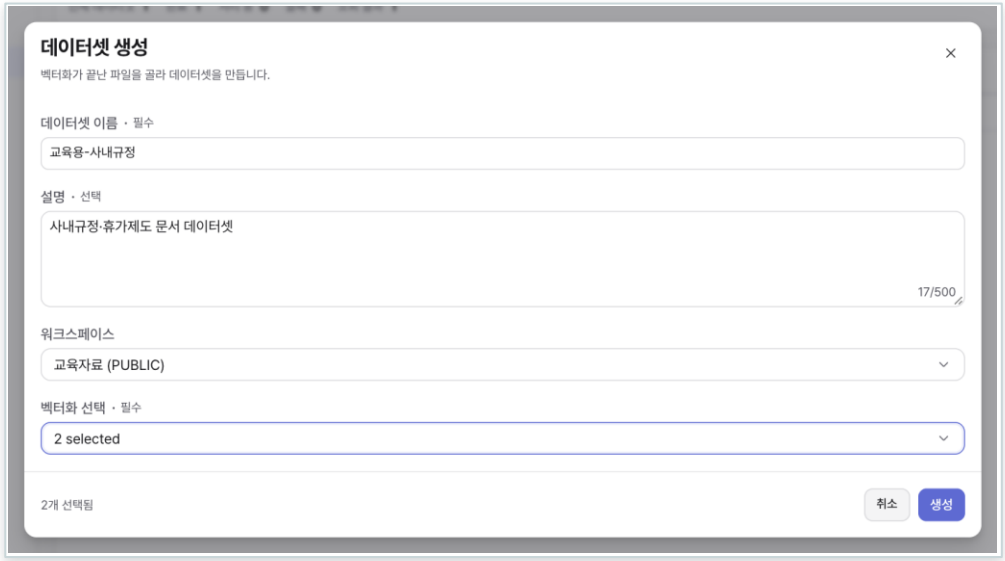
Data Fabric > 검색 데이터 > 데이터셋

1. **데이터셋 생성 창 열기**
   데이터셋 탭에서 [데이터셋 생성]

2. **이름과 설명 입력**
   예: 교육용-사내규정

3. **워크스페이스 선택**
   4-1에서 만든 워크스페이스

4. **벡터화 선택 후 [생성]**
   4-2에서 완료된 벡터화 선택




**참고** 훈련생은 지식 검색 노드에서 이 이름으로 데이터셋을 찾습니다. 알아보기 쉬운 이름을 씁니다.

# 준비 완료 체크리스트

- [ ] **계정**
  훈련생 계정 생성 (유형 관리자, 상태 활성), 계정표 준비
- [ ] **역할**
  '교육생' 역할에 훈련생 사용자와 그룹 할당
- [ ] **모델**
  gpt-4o, text-embedding-3-large 등록, 기본 모델 지정
- [ ] **데이터셋**
  벡터화 완료, 데이터셋 생성
- [ ] **사전 실행**
  강사 계정으로 Part 2 전 과정을 한 번 실행
- [ ] **훈련생 확인**
  훈련생 계정 하나로 로그인해 데이터셋이 보이는지 확인

# Part 2. 훈련생 실습

배부받은 계정으로 로그인해 나만의 챗봇과 문서 Q&A 앱을 만들고,
내 업무공간에서 직접 대화해 봅니다.

# 실습 전 확인사항

### 로그인

강사에게 받은 이메일과
비밀번호로 로그인합니다.

### 이름 규칙

모든 훈련생의 앱이 같은 목록에
보입니다.

워크플로 이름 앞에 본인 이름을
붙입니다.

예: 홍길동-기본챗봇

### 수정 금지

Administration과 검색 데이터
메뉴의 기존 항목(모델, 데이터셋,
역할)은 수정하거나 삭제하지
않습니다.

**참고** 훈련생 계정은 관리자 유형이라 강사가 준비한 항목도 바꿀 수 있습니다. 실습에 필요한 화면만 사용합니다.

# 5-1. 워크플로 만들기

Studios > 에이전트 빌더

1. **[워크플로우 생성]**
   에이전트 빌더 목록 화면 상단

2. **'워크플로우' 선택**
   '무엇을 만들까요?' 창에서 선택

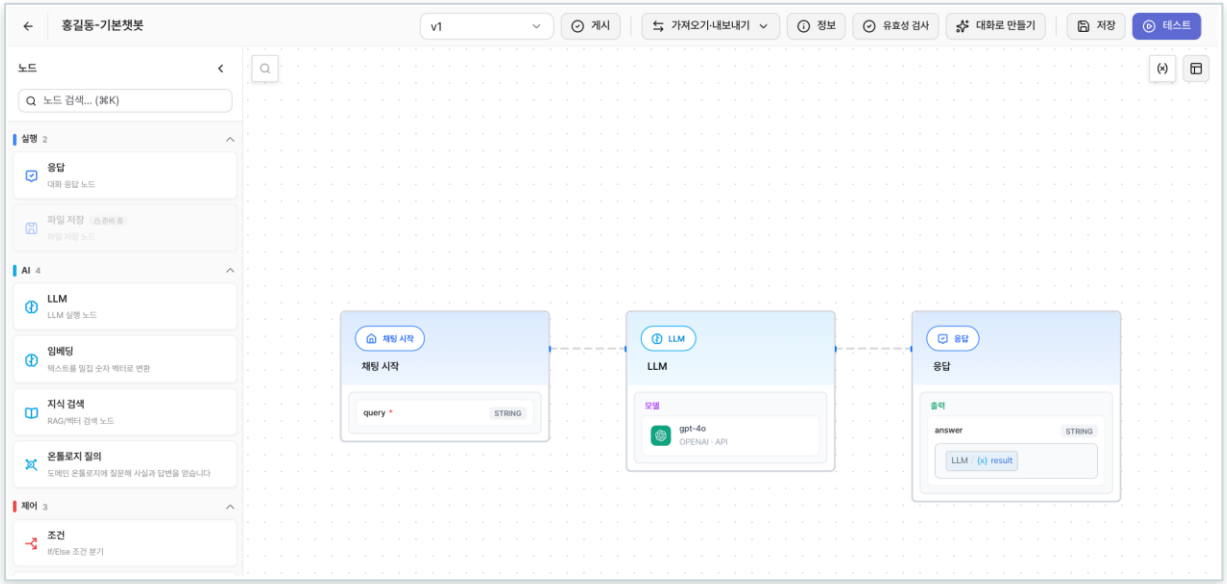
3. **이름, 설명, 유형 입력 후 [생성]**
   이름: 홍길동-기본챗봇 / 유형: 채팅

4. **편집 화면 열림**
   '채팅 시작' 노드가 자동으로 놓여 있음




**참고** 유형은 반드시 '채팅'을 고릅니다. API 유형은 대화 화면이 아니라 API 키로 호출하는 방식입니다.

# 5-2. 편집 화면 구성




**상단 툴바**
오른쪽 끝에 [저장], [테스트], 왼쪽에 [게시]가 있습니다.

**노드 목록 (왼쪽)**
LLM, 지식 검색, 응답 같은 노드를 캔버스에 추가합니다.

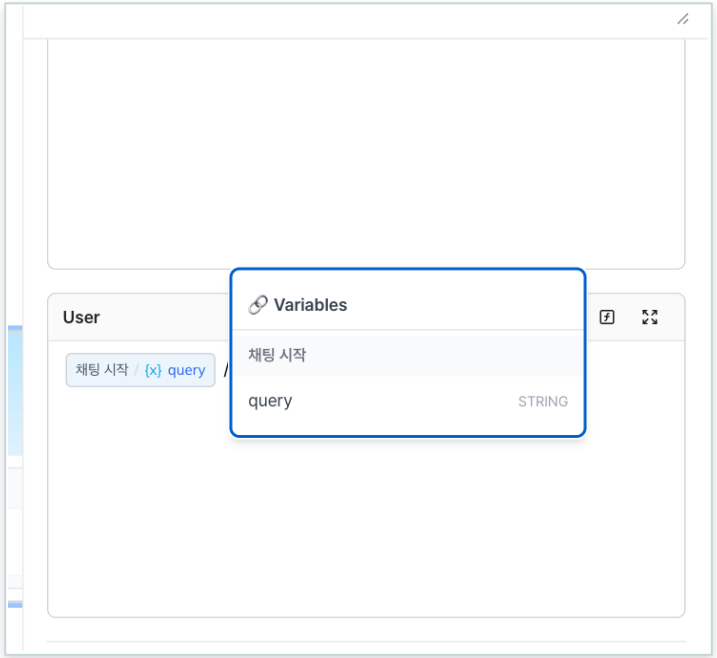
**캔버스 (가운데)**
노드를 선으로 연결해 처리 순서를 만듭니다. 노드를 누르면 오른쪽에 설정 패널이 열립니다.

**참고** '채팅 시작' 노드는 노드 목록에 없고 자동으로 만들어집니다. 지우지 않습니다.

# 5-3. 변수 넣기: / 입력

프롬프트(System, User)와 응답 입력란 공통

1. **변수를 넣을 위치에서 / 입력**
   'Variables' 목록이 열림
2. **변수 선택**
   앞 노드의 변수가 칩으로 들어감
3. **[변수 삽입] 버튼도 같은 기능**
   직접 타이핑하지 않습니다



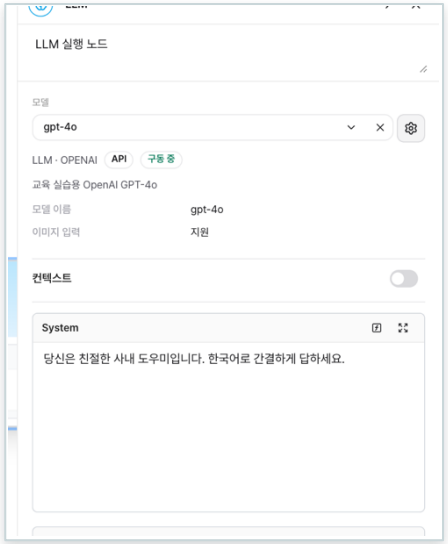

| 변수 칩 | 의미 |
|---|---|
| **[채팅 시작 / query]** | 사용자가 입력한 질문 |
| **[지식 검색 / result]** | 데이터셋에서 찾은 문서 |
| **[LLM / result]** | LLM이 만든 답변 |

**참고** 이 매뉴얼에서 대괄호 [ ]는 / 로 넣은 변수 칩을 뜻합니다. 목록에는 연결된 앞 노드의 변수만 나옵니다.

# 5-4. 예시 A: 기본 챗봇

홍길동-기본챗봇

**채팅 시작** → **LLM** → **응답**




## 설정할 값

LLM 노드

모델: gpt-4o

System: 당신은 친절한 사내 도우미입니다. 한국어로 간결하게 답하세요.

User: [채팅 시작 / query]

응답 노드

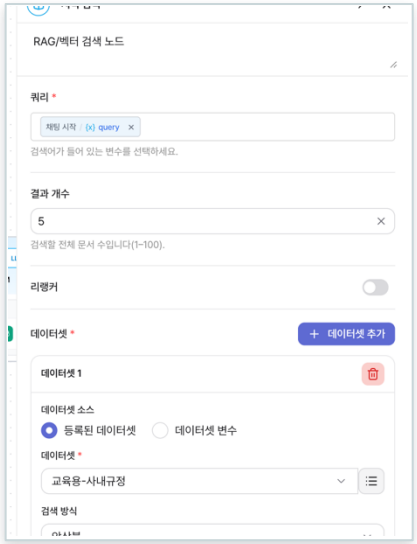
답변: [LLM / result]

**참고** 노드 연결 순서가 곧 처리 순서입니다. 응답 노드가 마지막에 연결되어 있는지 확인합니다.

# 5-5. 예시 B: 문서 Q&A

홍길동-문서QA (새 워크플로로 생성)

채팅 시작 → 지식 검색 → LLM → 응답




## 설정할 값

지식 검색 노드

쿼리: [채팅 시작 / query], 결과 개수: 5

[데이터셋 추가] → 교육용-사내규정

LLM 노드

System: 참고 문서만 근거로 답하세요. 문서에 없으면 '문서에서 찾을 수 없습니다'라고 답하세요.

User: 참고 문서 [지식 검색 / result] 질문 [채팅 시작 / query]

응답 노드: [LLM / result]

**참고** 데이터셋이 목록에 없으면 페이지를 새로고침합니다. 목록은 최대 5분 동안 이전 내용을 보여 줄 수 있습니다.

# 5-6. 테스트, 저장, 게시

편집 화면 상단 툴바

### 1 테스트

[테스트] → 질문 입력 → 실행
저장 전에도 실행됩니다.

### 2 저장

[저장]
게시는 저장한 뒤에만 가능합니다.

### 3 게시

[게시] → '쓸 수 있는 역할'에서
교육생 선택 → [게시]




**참고** 새 버전을 게시하면 이전 게시 버전은 자동으로 게시 해제됩니다. 예시 A와 B를 모두 게시합니다.

# 6. 역할에 워크플로 할당

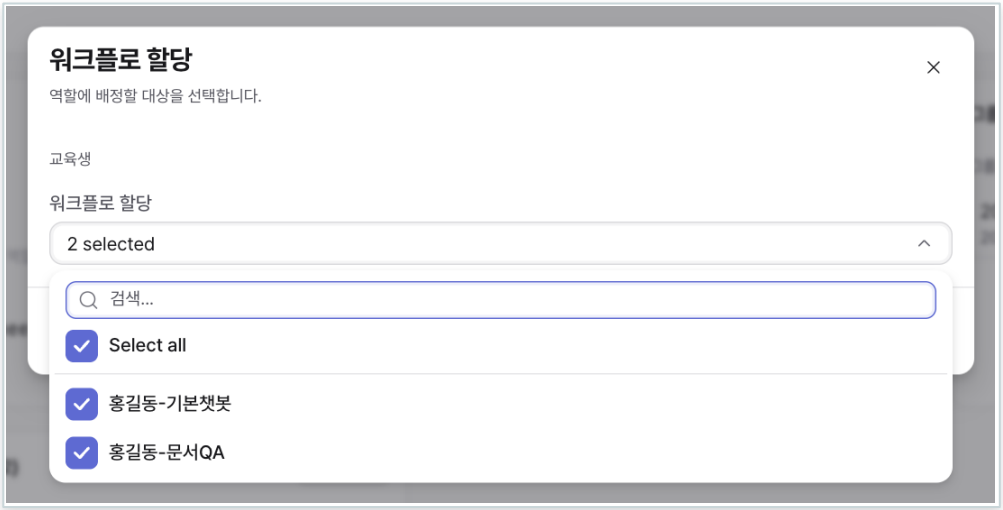
Administration > 역할 및 권한

1. **'교육생' 역할 열기**
   역할 목록에서 선택

2. **워크플로 카드의 [할당]**
   역할 상세 화면

3. **본인 워크플로 선택**
   홍길동-기본챗봇, 홍길동-문서QA

4. **이미 연결됐다면 확인만**
   게시할 때 교육생을 골랐다면 목록에 이미 있음




**참고** 할당 목록에는 게시하지 않은 워크플로도 보입니다. 게시한 본인 워크플로만 고릅니다.

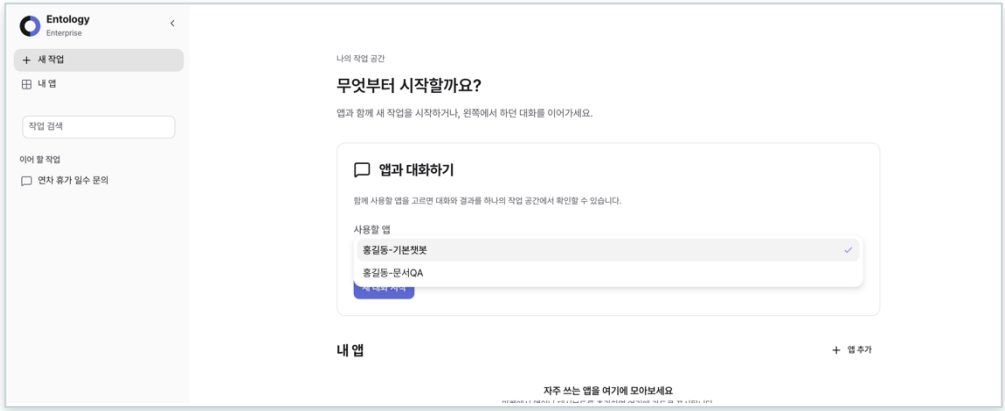
# 7-8. 내 업무공간 이동과 앱 선택

1. **[내 업무공간] 이동**
   화면 상단 메뉴

2. **사용할 앱 선택**
   '사용할 앱'에서 본인 앱 선택

3. **[새 대화 시작]**
   대화 화면이 열림

4. **(선택) 내 업무공간에 추가**
   서비스 마켓에서 [내 업무공간에 추가]




**참고** 추가하지 않아도 앱은 쓸 수 있습니다. 추가한 앱은 이 브라우저에만 저장됩니다.

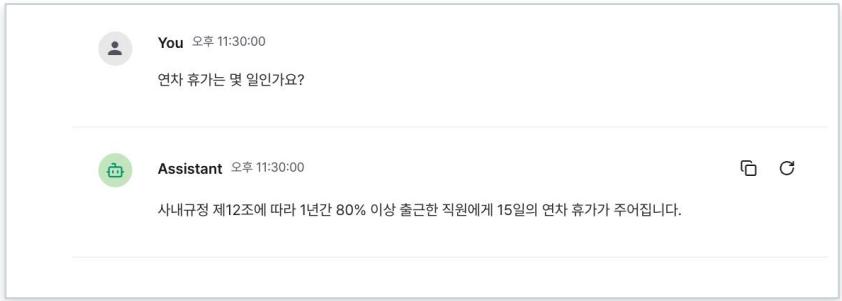
# 9. 대화하기

## 기본 챗봇 (예시 A)

일반 질문을 해 봅니다.
예: 회의록을 잘 쓰는 방법을 알려줘

## 문서 Q&A (예시 B)

업로드한 문서 내용을 물어봅니다.
문서에 없는 내용도 물어보고 답변 차이를 비교합니다.



# 문제 해결

| 증상 | 확인할 것 |
| --- | --- |
| **앱 목록에 내 워크플로가 없음** | 게시했는지, 게시할 때 '교육생' 역할을 골랐는지 |
| **지식 검색에 데이터셋이 없음** | 페이지 새로고침 후 다시 확인, 계속 없으면 강사에게 문의 |
| **답변 대신 오류가 남** | LLM 노드에 gpt-4o를 골랐는지, [테스트]에서 실패한 노드 확인 |
| **문서 Q&A가 문서와 다른 답을 함** | LLM의 User 프롬프트에 [지식 검색 / result]가 들어갔는지 |
| **[게시]가 눌리지 않음** | 먼저 [저장]했는지 |
| **(강사) 벡터화 [실행]이 비활성** | 파일과 임베딩 모델을 골랐는지, API 키가 맞는지 |

# 실습 완료

기본 챗봇과 문서 Q&A 앱을 만들고, 역할로 공유하고, 직접 대화해 봤습니다.