

HYPERTEAMS 4회차 · 85분

확장하기

두 개의 API 표면 · REST · 모델 갖추기 · 음성과 이미지

화면에서 쓰는 법이 아니라, 다른 프로그램이 이 컴퓨터를 부르는 법입니다

그래서 편의보다 안전이 먼저입니다 — 두 표면 모두 기본값은 꺼짐입니다

표면

바깥으로 열리는 문은 둘뿐

/api/v1

무엇을
업무를 시킨다

인증
Bearer

기본값
꺼짐

단계
켜짐 / 꺼짐

/api/ai

무엇을
✓ 모델을 부른다

인증
✓ Bearer

기본값
✓ 꺼짐

단계
✓ off · inference · full

그 밖의 모든 /api/* 는 대시보드 쿠키 전용 — 바깥에서 못 부릅니다.

모델 표면

얼마나 열지를 고릅니다

모드

OFF
기본값

INFERENCE
여기서 시작

FULL
관리까지

열리는 것

OFF
✓ 아무것도. 404

INFERENCE
✓ 추론만 — 대화 · 임베딩 · 음성 · 이미지

FULL
✓ 상류 모델 서버의 관리 경로까지

경로가 범위를 강제하지 않습니다. .../api/ai/v1 을 안내했다고 그 아래로만 갈 수 있는 게 아닙니다 — 범위를 정하는 것은 모드입니다.

열쇠가 곧 대시보드 비밀번호입니다

그래서 비밀번호가 없으면 켜지지 않고, 값이 복사되는 곳마다 위험이 늘어납니다

켜는 순서

한 곳에서 한 번 성공시키고 멈추세요

1

비밀번호

hyperteams setup — 비밀번호가 없으면 애초에 켤 수 없습니다

2

필요한 표면만

업무를 외부에서 시킬 것 → REST ·
모델을 빌려줄 것 → inference

3

열쇠 목록

그 값을 넣을 곳을 종이에 적습니다.
셀 수 없으면 켜지 마세요

4

한 곳만

한 번 성공하면 그다음은 복사입니다

처음부터 여러 곳에 붙이면 어디가 틀렸는지 못 찾습니다. 4번이 이 슬라이드의 핵심입니다.

REST

엔드포인트는 여섯 개가 전부입니다

01

GET /workspaces

작업 폴더 목록 — 여기서 번호를 얻습니다

02

GET /tasks

업무 목록

03

POST /tasks

업무 만들기

04

GET /tasks/{id}

업무 하나 보기

05

POST .../follow-up

이어 말하기

06

POST .../stop

종지

전부 /api/v1 아래입니다. 화면에서 하던 것과 같은 함수를 부르므로 결과도 화면에 그대로 보입니다.

REST

업무를 코드로 시키기

- POST /api/v1/tasks — 화면에서 만든 업무와 같은 함수
- follow-up 으로 이어 말하고 stop 으로 멈춤
- mode 를 빼면 전부 자동입니다
- workspaceId 는 하드코딩 말고 경로로 조회

역등 키가 없습니다 — 같은 요청을 두 번 보내면 업무가 두 개 생깁니다.

역등 키가 없습니다

중복 방지는 부르는 쪽 책임입니다

1

재시도 전에 조회

GET /api/v1/tasks 로 최근 목록을
먼저 확인

2

name 에 고유값

"일일점검-2026-08-15" — 이미
있는지 알아볼 수 있게

3

재시도 상한

무한 재시도 금지

4

파일을 고치는 업무면

재시도를 아예 사람 확인으로

타임아웃이 곧 실패는 아닙니다 — 요청은 도착했는데 응답만 못 받았을 수 있습니다. 그 상태의 재시도가 같은 일을 두 번 실행시킵니다.

모델

내 컴퓨터를 모델 서버로

- 대화 · 임베딩 · 음성 · 이미지가 같은 규격으로
- 나가면 안 되는 데이터를 로컬에서 처리
- 반복 · 대량 호출은 건당 비용이 없음
- 복제 목소리는 OpenAI 규격 밖 — 동의가 전제입니다

2줄

base 주소와 열쇠

OpenAI SDK 코드는 그대로

무엇을 내 컴퓨터로 돌릴 것인가

작업

높음

분류 · 태깅 · 임베딩

높음

음성 인식 · 합성

중간

긴 문서 추론

낮음

복잡한 코드 작성

판단

높음

✓ 반복 많고 난이도 낮음 · 대량 처리

높음

✓ 건당 비용 없이 길이만큼

중간

✓ 모델과 하드웨어에 달림

낮음

✓ 상용 모델이 앞섭니다

모델 라우팅의 로컬판입니다 — 난이도에 맞춰 고르되, 선택지에 “내 컴퓨터”가 하나 더 생긴 것입니다.

불일 때

실패하는 자리는 정해져 있습니다

01

모델 이름이 다릅니다

상용 API 이름을 그대로 쓰면 없습니다. /v1/models 로 먼저 목록 확인

02

응답 시간이 다릅니다

상용 기준 타임아웃이 그대로면 실패합니다. 한 번 재보고 여유를

03

508 이 나옵니다

주소가 자기 자신입니다. 버그가 아니라 무한 루프를 막는 장치

04

inference 인데 404

관리 경로를 부른 것입니다. 정말 필요하다면 full 이지만 권한이 커집니다

네 가지 모두 열쇠 문제가 아닙니다. 401 이 아닌 오류를 인증 문제로 오해하는 데서 시간이 가장 많이 셉니다.

모델 갖추기

받기 전에 이 PC에서 되는지부터

01

서버 — 둘 중 하나

이 컴퓨터에서 돌리기 / 이미 도는 주소에 붙기

02

프리셋 다섯

Whisper(음성→글) · Fish Speech(글→음성) · 대화 둘 · 이미지 하나

03

프리플라이트

메모리와 디스크를 따로 봅니다. “빠듯합니다”를 가볍게 보지 마세요

04

컨텍스트 = 메모리

8k는 1.3GiB, 128k는 21GiB. 무조건 크게가 아닙니다

엔진은 직접 고르지 않습니다. 설치 시점에 하드웨어에 맞춰 정해지므로 CPU 경고에는 '수리 다시 실행'입니다.

음성 · 이미지

오래 걸리고, 실패하면 부분만 다시 만듭니다

- 긴 원고는 문단으로 나눠 만드세요 — 실패한 문단만 다시 만들 수 있습니다
- 이미지 프롬프트는 파일로 — 재현 · 수정 · 일관성 셋이 여기서 나옵니다
- 낮은 해상도로 여러 장 → 고른 것만 확대(upscale)가 시간과 비용에 맞습니다
- 입력과 출력을 폴더로 나누고, 출력은 git에서 제외하세요

복제 목소리는 OpenAI 규격 밖입니다. 목소리는 개인정보이고 동의가 전제입니다 — 내부 자료여도 마찬가지입니다.

실습 · 40분

REST로 업무를 걸어보기

1

읽기부터

조회가 200 이면 토큰은 정상.
인증과 요청 문제를 분리합니다

2

읽기만 하는 업무

화면에도 같은 업무가 보이는지 —
두 표면이 같은 것을 봅니다

3

진행 읽기

조회 간격을 적어두세요. 자동화할
때 그대로 설계값이 됩니다

4

두 번 보내기

업무가 두 개 생깁니다 — 멍등
키가 없습니다

타임아웃은 실패가 아니라 응답이 늦은 것일 수 있습니다. 재시도는 실패가 확인됐을 때만 — 이게 가장 자주 사고를 냅니다.

정리

가져갈 것

- 1 문은 둘, 기본은 꺼짐, 열쇠는 대시보드 비밀번호
- 2 REST 에 먹등 키가 없다 — 재시도는 호출자 책임
- 3 OpenAI 호환이라 base 주소 한 줄로 갈아탄다
- 4 모델은 프리플라이트를 보고 작은 것부터 받는다
- 5 컨텍스트를 최대로 올리는 것은 메모리를 태우는 일이다
- 6 음성·이미지는 조각으로 만들고 프롬프트를 파일로 남긴다