

Vibe Coders

사용자 가이드

AI 코딩 도구를 게이트웨이에 연결하고 사용량을 본다

버전 v0.85.1

작성일 2026-09-12

문서 사용자 가이드

사용자 가이드

이 문서는 게이트웨이를 쓰는 사람(AI 코딩 도구를 연결하는 개발자, 자기 팀의 사용량을 보는 사람)을 위한 것입니다. 게이트웨이를 띄우고 지키는 방법은 [관리자 가이드](#)에 있습니다. PDF: [USER_GUIDE.pdf](#)

1. 이 제품이 하는 일

Vibe Coders 는 AI 코딩 도구(Roo Code, Cline, Cursor, Continue, OpenAI SDK 등)와 LLM 공급자 사이에 서는 **AI 프록시 게이트웨이**입니다. 도구의 Base URL 을 게이트웨이 주소로 바꾸면, 코드나 워크플로를 바꾸지 않아도 모든 호출이 게이트웨이를 지나며 사용량·비용·지연·언어 통계가 회사에 기록됩니다.

게이트웨이는 업스트림 API 키를 대신 보관하고, 개발자에게는 **proxy key** 하나만 줍니다. 프롬프트에 섞인 주민번호·카드번호·클라우드 키 같은 비밀은 자동으로 마스킹되고, 팀·키·IP 단위의 비용 한도와 정책이 호출 시점에 적용됩니다.

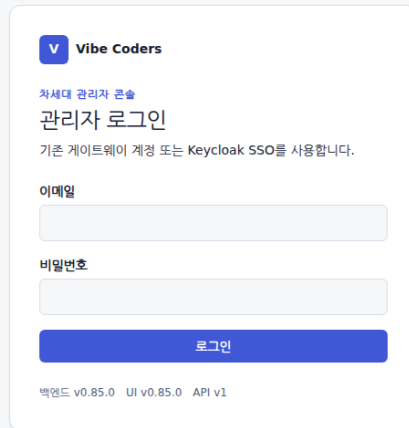
콘솔(/app)에서는 자기 사용량과 팀 대시보드, 요청 이력, 세션 비행기록을 볼 수 있습니다. 관리자 역할이면 같은 콘솔에서 사용자·정책·비용·보안 설정까지 다룹니다.

2. 처음 5분

로그인부터 첫 호출이 콘솔에 잡히는 것까지 한 번에 따라 합니다. 운영자에게 **게이트웨이 주소**(예: `http://proxy-gateway.intra:8080`), **콘솔 계정**(이메일·비밀번호 또는 SSO), **proxy key**(`pcg_...` — 한 번만 표시되니 받자마자 보관)를 받아 두세요.

2.1 콘솔에 로그인

브라우저에서 `http://<게이트웨이>/app` 을 엽니다. 운영자가 계정 로그인(`AUTH_ENABLED=true`)을 켜다면 이메일·비밀번호 폼이, 기존 관리자 토큰 방식이면 토큰 입력란이, SSO 를 켜다면 SSO 버튼이 보입니다.



Vibe Coders

차세대 관리자 콘솔

관리자 로그인

기존 게이트웨이 계정 또는 Keycloak SSO를 사용합니다.

이메일

비밀번호

로그인

백엔드 v0.85.0 UI v0.85.0 API v1

관리자 로그인 — 이메일·비밀번호로 콘솔에 들어간다

2.2 도구의 Base URL 을 바꾼다

OpenAI SDK / Roo Code / Cline / Cursor 모두 같은 패턴입니다.

항목	기존	변경
Base URL	https://api.openai.com/v1	http://proxy-gateway.intra:8080/v1
API Key	sk-... (OpenAI 발급)	pcg_... (회사 발급 proxy key)
모델명	gpt-4.1-mini 등	그대로 사용

도구별 화면 위치는 [4.1 도구별 연결 설정](#)에 있습니다. 터미널에서 바로 확인하려면:

```
curl -sS http://proxy-gateway.intra:8080/v1/chat/completions \
  -H "Authorization: Bearer pcg_xxxxxxx" -H "Content-Type: application/json" \
  -d '{ "model": "gpt-4.1-mini", "messages": [{ "role": "user", "content": "안녕" }] }'
```

응답 헤더의 X-Request-ID 가 이 호출의 요청 ID 입니다.

2.3 콘솔에서 내 호출을 찾는다

왼쪽 메뉴 개요 → 내 호출을 열면 방금 호출이 본인 사용량으로 잡혀 있습니다.

V

Vibe Coders

자세대 콘솔

개요

통합 현황

내 내 홈

팀 팀 대시보드

AI 게이트웨이

게 게이트웨이 상태

A AI 공급자

모 모델

C Chat 테스트

라우팅

라 라우팅

관측

요 요청 탐색기

추 추적 탐색기

세 세션 비행기록

X XView 실시간

L LLM 관측

진 진단 프로브

프롭트 및 평가

< 사이드바 접기

Vibe Coders / 개요 / 내 홈

Q 검색

HK

●정상

자동 갱신 끄

기존 화면

A

미리보기

내 홈

admin@example.com 님의 사용량, 요청 영수증, API 키와 연결 상태를 확인합니다.

기존 화면에서 열기

새로고침

내 홈

요청·영수증

Skill·추천

내 카·연결

오늘 요청

0

오늘 오류

0

이번 달 비용

₩0

30일 성공률

0.0%

절감 가능 금액

₩0

개인 위험 점수

0

내 액션 큐

지금 처리하면 좋은 항목입니다.

처리할 액션이 없습니다.
비용, 키 만료, 정책 위반 신호가 생기면 여기에 모입니다.

알림 센터

읽지 않은 알림 0건 중 중요 0건

새 알림이 없습니다.
비용 초과, 정책 위반, 키 만료 같은 사건이 생기면 알려 드립니다.

내 리포트

선택한 기간의 사용량과 직전 기간 대비 변화입니다.

기간

주간

요청

0

토큰

0

비용

₩0

성공률

0.0%

캐시 적중률

0.0%

직전 대비 비용

0.0%

감소

내 홈 — 오늘의 요청 수·토큰·비용과 최근 호출을 본다

관측 → 요청 탐색기에서 "요청 ID" 필터에 X-Request-ID 값을 넣고 조회를 누르면 그 호출 한 건의 상태·모델·지연·토큰·비용이 나옵니다.

V Vibe Coders

자세한 콘솔

Vibe Coders / 관측 / 요청 탐색기

Q 검색

⌘K

●정상

자동 갱신 끄

기존 화면

A

개요

통합 현황

읽기 전용

내 홈

미리보기

팀 대시보드

미리보기

AI 게이트웨이

게이트웨이 상태

미리보기

A AI 공급자

미리보기

모 모델

미리보기

C Chat 테스트

미리보기

라우팅

라우팅

미리보기

관측

요 요청 탐색기

읽기 전용

추적 탐색기

읽기 전용

세 세션 비행기록

미리보기

X XView 실시간

미리보기

L LLM 관측

미리보기

진 진단 프로브

미리보기

프롭트 및 평가

< 사이드바 접기

관측 · 읽기 전용 미리보기

요청 탐색기

프롭트와 원시 오류를 제외한 최근 요청 운영 메타데이터를 조회합니다.

↺ 새로고침

↺ 기존 화면 보기

▽ 조회 필터

필터와 페이지 위치는 URL에 저장됩니다.

시작 시각

종료 시각

상태

예: 2026-09-04T09:00 또는 RFC3339

예: 2026-09-04T18:00 또는 RFC3339

전체

모델

요청 ID

공급자 참조

▶ 고급 필터

Q 조회

필터 초기화

시각	상태	요청 ID	모델	공급자	지연	토큰	비용	
26. 9. 12. 오전 6:58:13	200	req_8c9466efce86a8afa31632a38...	qwen-coder	openai	487ms	218	0.086 KRW	상
26. 9. 12. 오전 6:58:12	200	req_9081d35216570649f611bc32b...	gpt-4.1	openai	533ms	279	2.749 KRW	상
26. 9. 12. 오전 6:58:12	200	req_6542d587dff9982978efc97a5...	claude-sonnet-4	openai	551ms	104	1.49 KRW	상
26. 9. 12. 오전 6:58:11	200	req_8f84f396308b693c19bb90c1f...	gpt-4.1-mini	openai	592ms	401	0.819 KRW	상
26. 9. 12. 오전 6:58:11	200	req_7f086f2ea1d81f79d03d69406...	qwen-coder	openai	314ms	211	0.083 KRW	상
26. 9. 12. 오전 6:58:11	200	req_1702f914221928b45a7ff4cb4...	gpt-4.1	openai	268ms	306	3.047 KRW	상

요청 탐색기 — 최근 요청을 시각·상태·모델·비용으로 본다

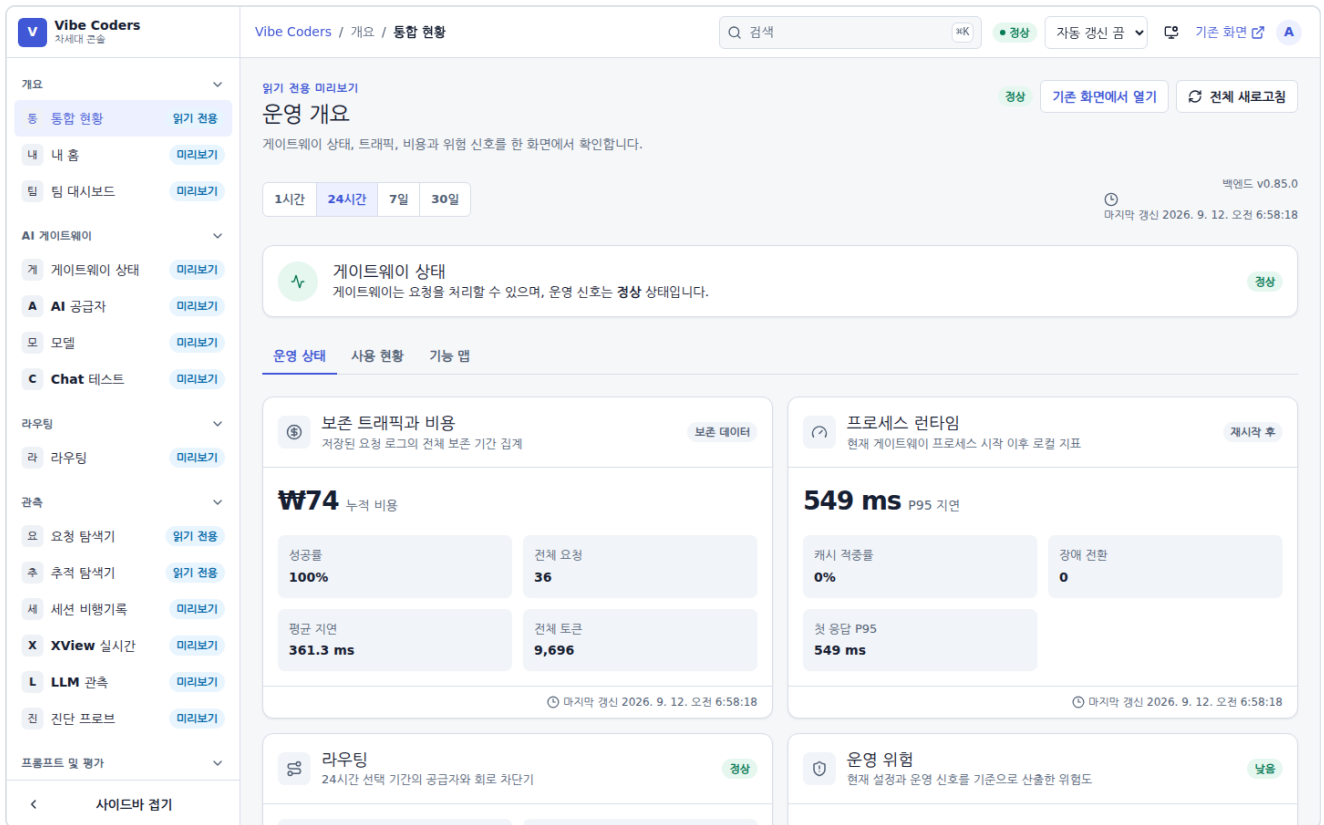
여기까지 되면 연결이 끝난 것입니다. 통계가 anonymous 나 passthrough 로 잡히면 등록되지 않은 키를 쓰고 있는 것이니 5.3을 보세요.

3. 화면별 사용법

메뉴 이름은 콘솔 왼쪽 사이드바의 문구 그대로입니다. 오른쪽 위 검색(⌘K / Ctrl+K)에 화면 이름을 치면 바로 이동합니다. 배지 읽기 전용은 보기만 되는 화면, 미리보기는 새 콘솔로 옮겨진 화면입니다. 기존 화면 링크는 같은 데이터를 예전 /admin 화면에서 엽니다.

3.1 통합 현황

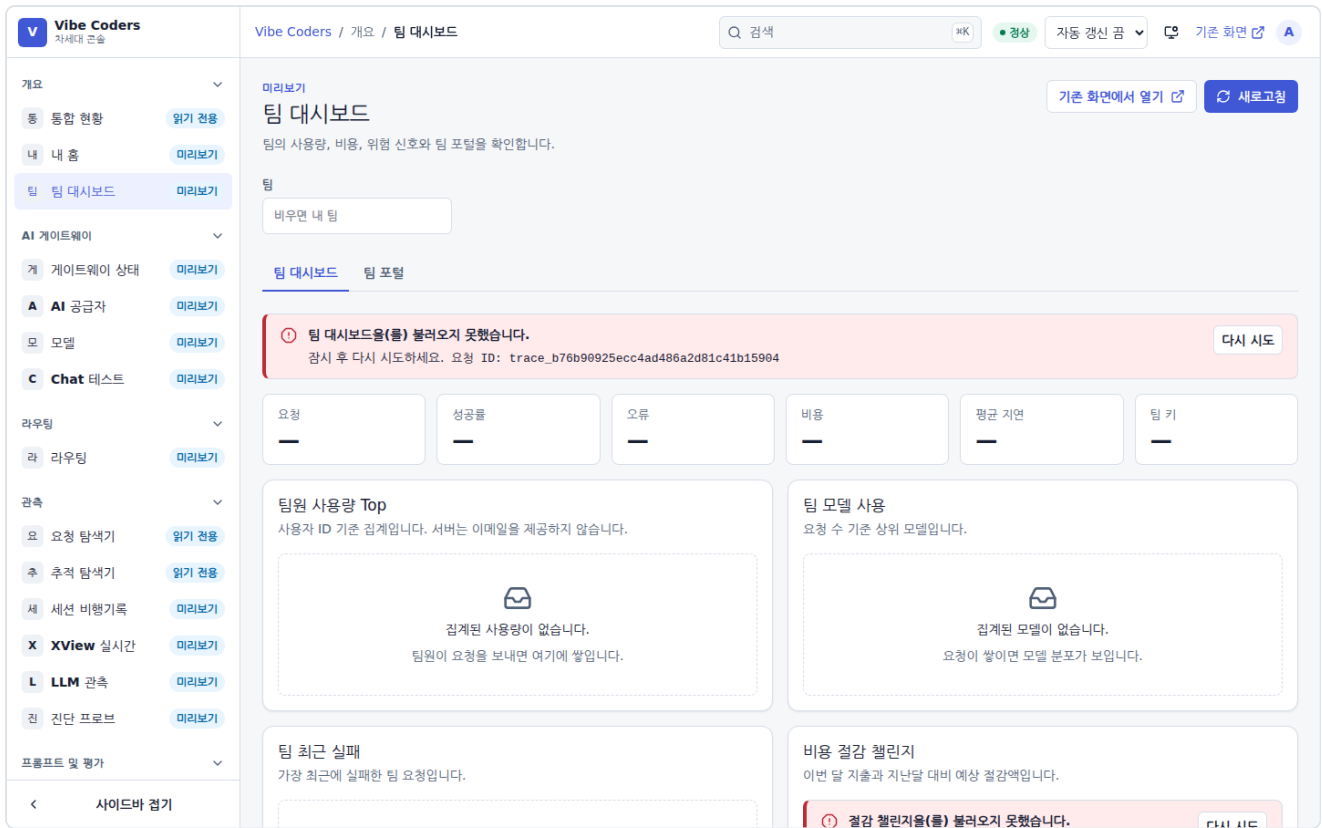
게이트웨이 상태, 보존 트래픽과 비용, 프로세스 런타임(P95 지연·캐시 적중률·장애 전환), 라우팅 상태, 운영 위험을 한 화면에서 봅니다. 기간은 1시간 / 24시간 / 7일 / 30일 버튼으로 바꿉니다. 탭 운영 상태 / 사용 현황 / 기능 맵 으로 관점을 바꿀 수 있습니다.



통합 현황 — 게이트웨이 상태·보존 비용·P95 지연을 한 화면에서 본다

3.2 내 홈 · 팀 대시보드

내 홈은 로그인한 본인의 요청·토큰·비용·최근 호출입니다. 팀 대시보드는 소속 팀의 합계와 팀원별 분포입니다. 월말 정산이나 "누가 얼마나 썼나"는 여기서 봅니다.



팀 대시보드 — 팀 합계와 팀원별 사용량 분포

3.3 요청 탐색기 · 추적 탐색기

요청 탐색기는 최근 요청의 운영 메타데이터(프롬프트 본문 제외)입니다. 시작·종료 시각, 상태, 모델, 요청 ID, 공급자 참조로 거르고 **고급 필터**를 펼치면 더 좁힐 수 있습니다. 필터와 페이지 위치는 URL 에 저장되므로 링크를 그대로 동료에게 보낼 수 있습니다. 행을 클릭하면 단건 상세가 열립니다.

추적 탐색기는 한 요청이 파이프라인의 어떤 단계(인증 → 정책 → 라우팅 → 업스트림 → 로깅)를 얼마 만에 지났는지 봅니다.

V Vibe Coders

자세대 콘솔

Vibe Coders / 관측 / 추적 탐색기

Q 검색

HK

정상

자동 갱신 끄

기본 화면

A

개요

종합 현황

읽기 전용

내 홈

미리보기

팀 팀 대시보드

미리보기

AI 게이트웨이

게 게이트웨이 상태

미리보기

A AI 공급자

미리보기

모 모델

미리보기

C Chat 테스트

미리보기

라우팅

라 라우팅

미리보기

관측

요 요청 탐색기

읽기 전용

추 추적 탐색기

읽기 전용

세 세션 비행기록

미리보기

X XView 실시간

미리보기

L LLM 관측

미리보기

진 진단 프로브

미리보기

프롭퍼트 및 평가

< 사이드바 접기

관측 · 읽기 전용 미리보기

읽기 전용

요청 탐색기

기본 화면 보기

새로고침

추적 탐색기

같은 추적 ID로 연결된 요청의 시작 시점과 처리 지연을 요청 단위로 확인합니다.

추적 조회

필터와 선택한 요청은 URL에 저장됩니다.

추적 ID

예: trace-01H...

상태

전체

모델

비워 두면 최근 요청의 추적 흐름을 표시합니다.

시작 시간

예: 2026-09-04T09:00 또는 RFC3339

종료 시간

예: 2026-09-04T18:00 또는 RFC3339

표시 건수

50건

시간대

Asia/Seoul

호출 조회

필터 초기화

추적 ID로 요청 흐름을 좁혀 보세요.

현재는 최근 요청을 표시합니다. 추적 ID를 입력하면 같은 추적에 속한 요청만 시간순으로 확인할 수 있습니다.

표시 요청

36건

추적 ID

36개

평균 지연

361ms

오류 요청

0건

요청 처리 흐름

각 막대는 안전한 운영 메타데이터로 구성된 요청 시작 시점과 전체 지연을 나타냅니다.

요청 단위

0ms

5.9초

11.8초

req_35ed7d8656c7ee0f0b4db0eaf188b23e gpt-4.1-mini

HTTP 200

추적 탐색기 — 요청 한 건의 파이프라인 단계별 소요 시간

3.4 세션 비행기록 · XView 실시간 · LLM 관측

- 세션 비행기록: 같은 X-Vibe-Session 으로 묶인 에이전트 세션을 시간순으로 되짚습니다. 도구 호출·재시도·비용 이 한 줄에 놓입니다.
- XView 실시간: 트랜잭션 응답시간 분포를 점으로 실시간 표시합니다. 점을 클릭하면 그 요청이 왜 그렇게 처리됐는지 설명이 뜹니다.
- LLM 관측: 모델별 토큰·지연·오류율과 평가 결과입니다.

V Vibe Coders
자세대 콘솔

개요

통 통합 현황 읽기 전용

내 내 홈 미리보기

팀 팀 대시보드 미리보기

AI 게이트웨이

게 게이트웨이 상태 미리보기

A AI 공급자 미리보기

모 모델 미리보기

C Chat 테스트 미리보기

라우팅

라 라우팅 미리보기

관측

요 요청 탐색기 읽기 전용

추 추적 탐색기 읽기 전용

세 세션 비행기록 미리보기

X XView 실시간 미리보기

L LLM 관측 미리보기

진 진단 프로브 미리보기

프롭프트 및 평가

< 사이드바 접기

Vibe Coders / 관측 / 세션 비행기록

Q 검색RK●정상자동 갱신 끄기기존 화면에서 열기 새로고침

미리보기

기본 화면에서 열기 새로고침

세션 비행기록

코딩 세션 단위로 게이트웨이 요청을 시간순으로 재구성해 원인을 추적합니다.

조회 기간(일) 7 세션 ID · 메시지 검색 세션 ID 또는 메시지 일부Q 조회초기화

세션 3
최근 7일

요청 36

오류 0

비용 ₩73.93

최근 세션
최근 코딩 세션(클라이언트 session_id)을 활동순으로 보여줍니다. 각 세션의 비행기록으로 드릴인하세요.

세션 ID	마지막 메시지	요청	오류	모델 수	토큰	비용	마지막 활동	
sess_c1d3f57258af	README 설치 걸을 다듬어줘	12	0	4	3,175	₩21.78	방금	상세
sess_d902a6a0908b1	이 SQL 쿼리에 인덱스를 추천해줘	12	0	4	3,327	₩29.07	방금	상세
sess_2485d4143726	테스트 케이스를 더 만들어줘	12	0	4	3,194	₩23.09	방금	상세

세션 비행기록 — 세션 단위로 묶인 호출을 시간순으로 본다

V Vibe Coders
자세대 콘솔

개요

통 통합 현황 읽기 전용

내 내 홈 미리보기

팀 팀 대시보드 미리보기

AI 게이트웨이

게 게이트웨이 상태 미리보기

A AI 공급자 미리보기

모 모델 미리보기

C Chat 테스트 미리보기

라우팅

라 라우팅 미리보기

관측

요 요청 탐색기 읽기 전용

추 추적 탐색기 읽기 전용

세 세션 비행기록 미리보기

X XView 실시간 미리보기

L LLM 관측 미리보기

진 진단 프로브 미리보기

프롭프트 및 평가

< 사이드바 접기

Vibe Coders / 관측 / XView 실시간

Q 검색RK●정상자동 갱신 끄기기존 화면에서 열기실시간지금 새로고침

미리보기

기본 화면에서 열기

XView 실시간

최근 요청의 분포와 이상치를 실시간 산점도로 관찰하고, 원인 세션까지 따라갑니다.

저장된 뷰
적점 설정 새로 저장 덮어쓰기 삭제 링크 복사

상태 조회 구간 시작 일시 종료 일시 검색 시간대
최근 1시간 mm/dd/yyyy, --:-- mm/dd/yyyy, --:-- Asia/Seoul

세로축 지표 축 스케일 점 구분 방식 모델 (필요 구분)
응답 지연(ms) 로그 상태별 gpt-4.1,gpt-4.1-mini

엔드포인트
/v1/chat/completions

필터 적용 초기화

분석 요청 36

오류 0

플백 0

정책 신호 0

지연 P95 551ms

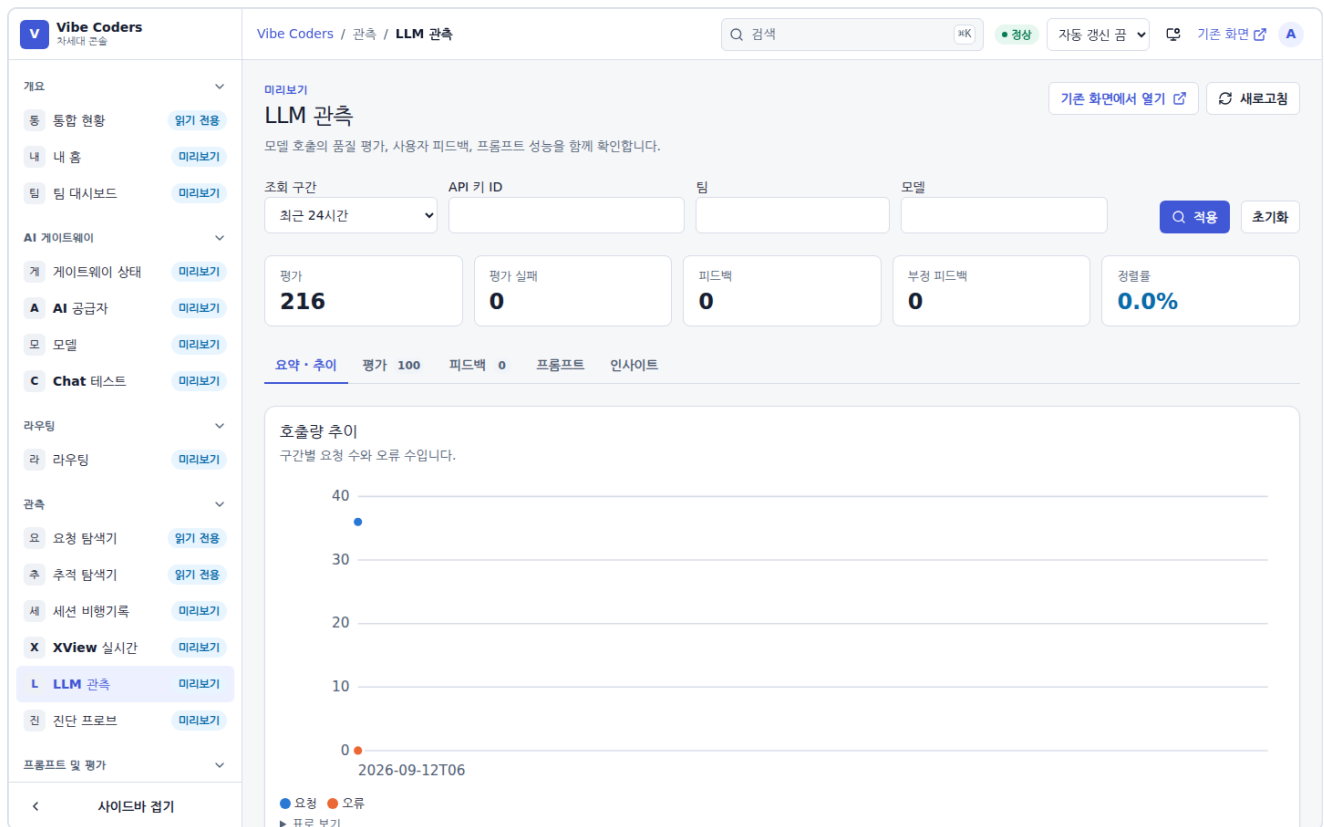
1.5초마다 갱신 마지막 갱신 방금

실시간 산점도 모델 집계 세션 워터폴

요청 분포
가로축은 시간, 세로축은 선택한 지표입니다. 그래프를 드래그하면 그 구간의 요청이 선택됩니다.

최근 25건 선택

XView 실시간 — 응답시간 분포를 점으로 본다



LLM 관측 — 모델별 토큰·지연·오류율

3.5 Chat 테스트 · 모델

Chat 테스트는 콘솔 안에서 모델에 바로 말을 걸어 보는 화면입니다. 도구 설정이 맞는지 의심될 때 게이트웨이 자체는 정상인지 여기서 먼저 가릅니다. 모델은 게이트웨이가 노출하는 모델 목록과 각 모델의 공급자·가격입니다.

V

Vibe Coders

차세대 콘솔

개요

통합 현황

내 홈

팀 대시보드

AI 게이트웨이

게이트웨이 상태

AI 공급자

모델

Chat 테스트

라우팅

라우팅

관측

요청 탐색기

추적 탐색기

세션 비행기록

XView 실시간

LLM 관측

진단 프로브

프롬프트 및 평가

사이드바 접기

Vibe Coders / AI 게이트웨이 / Chat 테스트

Q 검색

HK

정상

자동 갱신 끄

기존 화면 열기

A

미리보기

Chat 테스트

기존 화면에서 열기

실제 호출입니다.
여기서 보내는 요청은 실제 공급자로 나가며 비용이 발생합니다. 프롬프트와 응답은 이 화면에만 남고 주소나 브라우저 저장소에 저장되지 않습니다.

단일 호출멀티 모델 비교리더보드모델 용도 태그

호출 설정

게이트웨이를 통해 모델을 직접 호출합니다. 프롬프트와 응답은 화면에만 남고 주소나 브라우저 저장소에 기록되지 않습니다.

테스트 대상

직접 입력

모델

vibe/auto

공급자

비우면 라우터가 선택합니다.

최대 토큰

4096

Temperature

0

API Key ID

해당 키의 정책으로 호출합니다. 키 원문이 아닌 식별자입니다.

Proxy Bearer 토큰

임력 전용입니다. 저장하거나 다시 표시하지 않습니다.

라우팅 우회 (X-Proxy-No-Route)

지능형 라우팅을 건너뛰고 입력한 모델을 그대로 호출합니다.

프롬프트

Reply with pong in one short sentence.

Chat 호출라우팅 미리보기대화 저우기

Chat 테스트 — 콘솔에서 모델에 바로 프롬프트를 보내 본다

V

Vibe Coders

차세대 콘솔

개요

통합 현황

내 홈

팀 대시보드

AI 게이트웨이

게이트웨이 상태

AI 공급자

모델

Chat 테스트

라우팅

라우팅

관측

요청 탐색기

추적 탐색기

세션 비행기록

XView 실시간

LLM 관측

진단 프로브

프롬프트 및 평가

사이드바 접기

Vibe Coders / AI 게이트웨이 / 모델

Q 검색

HK

정상

자동 갱신 끄

기존 화면 열기

A

읽기 전용 미리보기

읽기 전용

기존 화면에서 열기

새로고침

모델

공급자별 모델 재고와 모델 ID 기준 품질, 가격, 사용 지침을 함께 조회합니다.

전체 모델

4

사용 가능

4

가상

0

확인 필요

0

모델 카탈로그

모델 계약

지원 종료

모델 검색

Q 모델 ID, 공급자, 소유자, 사용 지침

검색

공급자

전체 공급자

상태

전체 상태

품질 기간

1시간

24시간

7일

30일

필터 초기화

모델 목록

4개 결과 · 공급자와 모델 ID 기준으로 표시합니다.

마지막 갱신 2026. 9. 12. 오전 6:58:23

모델	공급자	상태	품질	성공률	입력 / 100만 토큰	출력 / 100만 토큰	적합한 용도
claude-sonnet-4 openai	openai 실시간	사용 가능	100점	100%	₩4,140	₩20,700	- 상세
gpt-4.1 openai	openai 실시간	사용 가능	100점	100%	₩2,760	₩11,040	- 상세
gpt-4.1-mini openai	openai 실시간	사용 가능	100점	100%	₩552	₩2,208	- 상세
qwen-coder openai	openai 실시간	사용 가능	100점	100%	₩150	₩450	- 상세

모델 — 사용 가능한 모델과 공급자·단가 목록

3.6 프롬프트 라이브러리

팀이 공유하는 프롬프트 템플릿과 골든 프롬프트(회귀 검사용)입니다. 반복해서 쓰는 지시문은 여기 올려 두고 X-Vibe-Knowledge 헤더로 끌어 씁니다(4.1 의 Knowledge Cache).

V Vibe Coders
자세한 내용

개요

- 동 통합 현황 [읽기 전용](#)
- 내 내 홈 [미리보기](#)
- 팀 팀 대시보드 [미리보기](#)

AI 게이트웨이

- 게 게이트웨이 상태 [미리보기](#)
- A AI 공급자 [미리보기](#)
- 모 모델 [미리보기](#)
- C Chat 테스트 [미리보기](#)

라우팅

- 라 라우팅 [미리보기](#)

관측

- 요 요청 탐색기 [읽기 전용](#)
- 추 추적 탐색기 [읽기 전용](#)
- 세 세션 비행기록 [미리보기](#)
- X XView 실시간 [미리보기](#)
- L LLM 관측 [미리보기](#)
- 진 진단 프로브 [미리보기](#)

프롬프트 및 평가

- < 사이드바 접기

Vibe Coders / 프롬프트 및 평가 / 프롬프트 라이브러리

Q 검색 🔍 ● 정상 자동 갱신 끄 🔄 기존 화면 [↩](#) A

미리보기

프롬프트 라이브러리

프롬프트 검색과 지문, 재사용 자산, 프롬프트 부채를 한곳에서 관리합니다.

[프롬프트 검색](#) [자산 관리소](#) [프롬프트 부채](#)

프롬프트 검색

키워드(#태그 포함)와 조건으로 프롬프트가 포함된 호출을 찾습니다. 프롬프트 원문은 마스킹된 형태로만 표시됩니다.

키워드

언어

클라이언트 IP

API 키 ID

이후 시각

예: refactor 또는 #security

예: java

mm/dd/yyyy, --:-- --

최대 건수

100

🔍 검색

📄 CSV 다운로드

저장된 필터

불러올 필터를 선택하세요

저장할 이름

예: 보안 리뷰 프롬프트

🗄 현재 필터 저장

시각	요청	모델	상태	지연(ms)	토큰	비용	프롬프트(마스킹)
2026-09-12 06:58:13	req_8c9466ef...	qwen-coder	200	487	218	₩0.09	You are a coding assistant.
2026-09-12 06:58:12	req_9081d352...	gpt-4.1	200	533	279	₩2.75	You are a coding assistant.
2026-09-12 06:58:12	req_6542d587...	claude-sonnet-4	200	551	104	₩1.49	You are a coding assistant.
2026-09-12 06:58:11	req_8f84f396...	gpt-4.1-mini	200	592	401	₩0.82	You are a coding assistant.
2026-09-12 06:58:11	req_7f686f2e...	qwen-coder	200	314	211	₩0.08	You are a coding assistant.
2026-09-12 06:58:11	req_1762f914...	gpt-4.1	200	268	306	₩3.05	You are a coding assistant.

프롬프트 라이브러리 — 공유 프롬프트와 골든 프롬프트 목록

4. 자주 하는 작업

4.1 도구별 연결 설정

4.1.1 Roo Code (VS Code 확장)

- VS Code 설정에서 Roo Code: OpenAI Base URL 검색
- http://proxy-gateway.intra:8080/v1 입력
- Roo Code: OpenAI API Key 에 pcg_xxxxxxx... 입력
- 모델을 평소 쓰던 것 (gpt-4.1-mini 등) 으로 선택

4.1.2 Cline

- 설정 → API Provider 를 OpenAI Compatible 로 선택
- Base URL: http://proxy-gateway.intra:8080/v1
- API Key: pcg_...
- Model: 원하는 모델명 (gpt-4.1-mini , claude-3-5-sonnet 등)

4.1.3 Cursor

1. Cmd/Ctrl + , → Cursor Settings → Models
2. "Add Custom OpenAI Base URL" 토글
3. URL: `http://proxy-gateway.intra:8080/v1`
4. API Key: `pcg_...`

4.1.4 Continue (VS Code/JetBrains)

`~/.continue/config.json` 의 model 에 다음 추가:

```
{
  "models": [
    {
      "title": "회사 프록시",
      "provider": "openai",
      "model": "gpt-4.1-mini",
      "apiBase": "http://proxy-gateway.intra:8080/v1",
      "apiKey": "pcg_XXXXXXX..."
    }
  ]
}
```

4.1.5 OpenAI Python SDK

```
from openai import OpenAI

client = OpenAI(
    base_url="http://proxy-gateway.intra:8080/v1",
    api_key="pcg_XXXXXXX...",
)

resp = client.chat.completions.create(
    model="gpt-4.1-mini",
    messages=[{"role": "user", "content": "main.go 를 리팩터링해줘"}],
)

print(resp.choices[0].message.content)
```

4.1.6 OpenAI Node SDK

```
import OpenAI from "openai";

const client = new OpenAI({
  baseURL: "http://proxy-gateway.intra:8080/v1",
  apiKey: "pcg_XXXXXXXX...",
});

const resp = await client.chat.completions.create({
  model: "gpt-4.1-mini",
  messages: [{ role: "user", content: "src/foo.ts 검토" }],
});

console.log(resp.choices[0].message.content);
```

4.1.7 curl

```
curl http://proxy-gateway.intra:8080/v1/chat/completions \
-H "Authorization: Bearer pcg_XXXXXXXX..." \
-H "Content-Type: application/json" \
-d '{
  "model": "gpt-4.1-mini",
  "stream": true,
  "messages": [{"role": "user", "content": "hello"}]
}'
```

`stream=true` 도 일반 OpenAI 응답과 동일하게 SSE 로 즉시 흘러나옵니다(게이트웨이가 버퍼링하지 않음).

4.1.8 선택: LLM 관측 메타데이터

운영자가 세션별 비용, 프롬프트 버전별 품질, 평가 실패를 추적해야 한다면 클라이언트에서 다음 헤더를 추가할 수 있습니다.

```
X-LLM-Session-ID: sess-123
X-LLM-Prompt-Name: code-review
X-LLM-Prompt-Version: v7
X-LLM-Prompt-Variables-Hash: vars-sha256
```

헤더가 없어도 호출은 정상 처리됩니다. prompt 메타데이터가 없으면 prompt는 `ad-hoc` 으로 표시됩니다. session은 아래 규칙으로 자동 그룹화됩니다.

세션 그룹화 — 무엇을 보내면 되나

게이트웨이는 명시적 → 추론 순으로 세션을 정합니다.

- 세션을 보내는 경우(권장): 다음 중 아무거나. 헤더가 바디보다 우선합니다.
 - 헤더: X-Session-ID, X-Vibe-Session-ID, X-Conversation-ID
 - 바디 필드: session_id (Langflow), chat_id (OpenWebUI), conversation_id, thread_id, 또는 metadata.session_id
- 세션을 안 보내는 경우(Claude Code·Cursor·Roo·Qwen 등 대부분의 코딩 툴): 게이트웨이가 api_key + IP + User-Agent 신원과 30분 비활성 윈도우로 세션(sess_...)을 자동 추론합니다. 한 작업 흐름의 연속 호출이 자연스럽게 한 세션으로 묶입니다. 30분 이상 멈췄다가 다시 호출하면 새 세션이 됩니다.

repo/branch 단위로 더 잘게 나누고 싶으면 X-Vibe-Repo · X-Vibe-Branch 헤더를 추가하세요(추론 신원에 반영됨). 한 작업을 확실히 한 세션으로 고정하려면 작업 시작 시 만든 UUID를 매 호출에 X-Vibe-Session-ID 로 보내는 것이 가장 정확합니다.

커밋/MR 과 세션 연결 (Prompt → Commit → MR)

프롬프트가 어떤 커밋·MR 로 이어졌는지 추적하려면, 커밋 메시지나 MR 제목에 세션 마커를 넣으세요. 운영자가 GitLab/Bitbucket 웹훅을 게이트웨이에 연결해 두었다면 자동으로 세션·사용자에 연결됩니다.

```
refactor OrderController
```

```
Vibe-Session: sess_8f34ab29 # 또는 [vibe:sess_8f34ab29]
```

commit-msg git 훅이나 커밋 템플릿으로 현재 세션 ID 를 자동 삽입하면 편리합니다. 세션 ID 는 /v1 응답을 직접 못 보는 도구라면 운영자에게 문의하거나, 직접 X-Vibe-Session-ID 로 지정한 값을 그대로 쓰면 됩니다.

4.1.9 MCP / 도구 사용 가시성

MCP 서버나 function calling 을 쓰는 경우(예: tools 배열을 보내거나 tool_calls 가 오가는 경우), 게이트웨이가 자동으로 어떤 서버·도구가 호출·실패했는지 집계합니다. 별도 설정은 필요 없습니다. mcp__<서버>__<도구> 형태의 도구 이름은 서버별로 자동 분류됩니다. 도구 결과(role:tool)가 오류({"isError":true} 등)이면 어드민 MCP 탭에서 오류로 집계되고, 운영자가 tool_error_rate 알림을 걸어두었다면 임계치 초과 시 통보됩니다.

4.1.10 Knowledge Cache — 반복 규칙을 짧게 참조하기

매번 같은 코딩 규칙·시스템 프롬프트를 통째로 보내는 대신, 운영자가 등록한 지식을 ID로 참조할 수 있습니다. 게이트웨이가 업스트림 전송 시 전체 본문으로 확장합니다(모델은 전체 텍스트를 받습니다).

```
# 방법 1) 메시지 본문 안에 플레이스홀더
{ "model": "gpt-4.1", "messages": [
  { "role": "user", "content": "{{kb: coding-standards}}\n\n위 규칙에 맞게 main.go 리팩터링" }
]}
```

```
# 방법 2) 헤더로 지식 주입 (시스템 메시지로 맨 앞에 추가됨)
curl http://proxy-gateway.intra:8080/v1/chat/completions \
  -H "Authorization: Bearer pcg_..." \
  -H "X-Vibe-Knowledge: coding-standards, security-rules" \
  -H "Content-Type: application/json" \
  -d '{ "model": "gpt-4.1", "messages": [{ "role": "user", "content": "main.go 리팩터링" }] }'
```

- 사용 가능한 ID는 운영자에게 문의하세요(설정 탭에 등록). 등록 안 된/중지된 ID는 확장되지 않고 플레이스홀더가 그대로 남습니다.
- 확장 여부는 응답 헤더 `X-Knowledge-Expanded: <id,...>` 로 확인할 수 있습니다.
- 장점: 규칙이 바뀌어도 클라이언트 수정 없이 자동 반영, 매 호출 본문이 짧아짐.

4.1.10b 비용 예측 헤더 / 큰 호출 승인

모든 chat 응답에는 게이트웨이가 호출 전에 추정된 값이 헤더로 붙습니다: `X-Estimated-Input-Tokens`, `X-Estimated-Output-Tokens`, `X-Estimated-Cost-KRW`, `X-Estimated-Latency-MS`. (`X-API-Key-Id` 로 어떤 키로 인식됐는지도 확인할 수 있습니다.)

운영자가 **비용 가드**를 켜 둔 경우, 예상 비용이 임계값을 넘는 호출은 `HTTP 402` 로 거부됩니다. 의도한 대형 작업이면 같은 요청에 `X-Cost-Approve: 1` 헤더를 붙여 다시 보내면 승인되어 통과합니다.

```
curl http://proxy-gateway.intra:8080/v1/chat/completions \
  -H "Authorization: Bearer pcg_..." -H "X-Cost-Approve: 1" \
  -H "Content-Type: application/json" -d '{ "model": "...", "messages": [...] }'
```

4.1.11 MCP Gateway — 여러 MCP 서버를 한 곳에 연결

여러 MCP 서버(GitHub·파일시스템·사내 도구 등)를 각각 등록하는 대신, 게이트웨이 한 곳만 클라이언트에 설정하면 등록된 모든 서버의 도구를 함께 쓸 수 있습니다.

- 클라이언트(Claude Code·Cursor 등)의 MCP 서버 URL 을 `http://<gateway>:8080/mcp` 로 설정.
- 인증은 LLM 호출과 동일하게 `Authorization: Bearer pcg_...` (proxy key).
- 도구·프롬프트 이름은 `<업스트림ID>_<이름>` 형태로 보입니다(예: `github__create_issue`). 리소스는 원본 URI 그대로 보입니다. 운영자가 어떤 업스트림을 등록했는지는 운영자에게 문의하세요.
- 지원: `initialize` / `tools/list` · `tools/call` / `resources/list` · `resources/read` · `resources/templates/list` / `prompts/list` · `prompts/get` / `ping` (JSON-RPC 2.0, Streamable HTTP). 도구·프롬프트·리소스 세 가지를 모두 집약합니다.

업스트림 등록·정책(차단/allowlist)은 운영자가 어드민 MCP 탭에서 관리하며, 게이트웨이를 통한 모든 도구 호출은 사용량·오류·반복 호출(루프) 관측에 자동 집계됩니다.

4.1.12 /mcp vs /mcp/gateway — 두 엔드포인트 구분

게이트웨이에는 이름이 비슷한 두 가지 MCP 엔드포인트가 있습니다. 용도가 다릅니다.

- **/mcp (업스트림 집약)**: 위 3.11 처럼 운영자가 등록한 외부 MCP 서버들의 도구를 한 곳에 모아 씁니다. 도구 이름은 <업스트림ID>_<이름>.
- **/mcp/gateway (게이트웨이 자체 기능)**: 게이트웨이 자신의 기능(chat·라우팅 미리보기·사용량/쿼터 조회·Text2SQL 미리보기·앱/워크플로 실행 등)을 MCP 도구로 노출합니다. 도구 이름은 gateway_chat · gateway_route_preview · gateway_get_usage_summary · gateway_run_workflow 등. 업스트림 등록이 필요 없습니다.

별도 SDK 없이 Claude Desktop·Cursor·Roo·Cline 같은 MCP 클라이언트에서 게이트웨이 기능을 바로 쓰려면 /mcp/gateway 를 설정하세요. 두 엔드포인트 모두 같은 proxy key 로 인증하며 본인 권한·쿼터·정책이 그대로 적용됩니다.

```
{ "mcpServers": { "vibe-gateway": {
  "url": "http://<gateway>:8080/mcp/gateway",
  "headers": { "Authorization": "Bearer pcg_..." }
} } }
```

연결이 잘 안 되면 내 홈 → "내 개발도구 연결하기 (MCP)" 카드에서 클라이언트를 고르고 연결 진단 버튼으로 인증·scope·모델 허용·쿼터·/v1/models·/mcp/gateway 도달성을 한 번에 점검할 수 있습니다(CLI 는 vibe doctor --client cursor). 설정 JSON 은 vibe mcp config 로도 출력됩니다.

4.2 provider 명시적 선택

회사가 여러 vendor 를 운영하는 경우 게이트웨이가 자동으로 적절한 곳으로 라우팅합니다.

- model=claude-3-5-sonnet → anthropic 자동 라우팅
- model=gpt-4.1-mini → openai 기본 라우팅

수동으로 강제하려면 X-Proxy-Provider 헤더를 추가하면 됩니다.

```
curl http://proxy-gateway.intra:8080/v1/chat/completions \
-H "Authorization: Bearer pcg_..." \
-H "X-Proxy-Provider: openrouter" \
-H "Content-Type: application/json" \
-d '{"model": "openai/gpt-4.1-mini", "messages": [...]}'
```

OpenAI SDK 처럼 헤더를 직접 못 넣는 클라이언트라면 운영자에게 "openrouter 로 모델 패턴 등록" 을 요청하세요.

4.3 비용/쿼터 한도

회사 정책에 따라 API 키 / 팀 / IP 단위로 일별·월별 한도가 걸려 있을 수 있습니다. 한도를 초과하면 호출은 다음과 같이 응답합니다.

```
HTTP/1.1 429 Too Many Requests
Retry-After: 1234
X-Quota-Scope: api_key:key_xxxxxxx:daily
X-Quota-Tokens: 950000
X-Quota-Cost-KRW: 49850.00
X-Quota-Period-Start: 2026-06-02T00:00:00+09:00
X-Quota-Period-End: 2026-06-03T00:00:00+09:00

{"error":{"message":"quota exceeded: krw_limit_exceeded", ...}}
```

`Retry-After` 는 다음 기간 시작까지의 초입니다. 한도가 늘어나야 한다면 운영자에게 요청하세요.

4.4 마스킹 / 프라이버시

게이트웨이는 다음 패턴을 프롬프트/응답에서 자동 마스킹합니다.

- 한국 주민번호 / 휴대전화 / 사업자등록번호
- 카드번호 (13~19자리)
- 이메일, 공인 IPv4
- AWS access key, GitHub/Slack 토큰, Google API key
- OpenAI `sk-...`, Anthropic `sk-ant-...`
- JWT, PEM private key
- `api_key=...`, Bearer `...` 형태 일반 시크릿

마스킹 텍스트는 `[REDACTED_RRN]`, `[REDACTED_OPENAI_KEY]` 처럼 라벨이 붙어 어드민에서 어떤 종류였는지 확인할 수 있습니다. 원문은 기본적으로 저장되지 않습니다(운영 정책에 따라 `LOG_RAW_PROMPTS=true` 일 때만 저장).

코드 컨텍스트에 비밀이 섞여 있는 경우, 마스킹 라벨이 본문에 들어가서 결과가 약간 어색할 수 있습니다. AI 코딩 도구에 비밀을 직접 붙여넣지 않는 게 가장 안전합니다.

4.5 한 줄 점검

```
curl -fsS http://proxy-gateway.intra:8080/v1/models | head
```

200 + 모델 리스트 JSON 이 오면 게이트웨이와 upstream 연결이 정상입니다. 모델 목록 조회는 SDK 호환성을 위해 인증 없이 허용됩니다. 실제 채팅/임베딩 호출에서 401 이 나오면 키가 잘못되었거나 비활성화된 것입니다.

5. 막혔을 때

5.1 호출이 오류로 돌아온다

응답 본문은 OpenAI 형식 `{"error":{"message":"...", "type":"...", "code":"..."}}` 입니다. `code` 로 찾으세요.

HTTP	code	message	무엇을 하면 되나
401	<code>invalid_api_key</code>	<code>invalid proxy API key</code>	도구에 넣은 키가 <code>pcg_...</code> proxy key 인지, 앞뒤 공백이 없는지 확인. 비활성화됐을 수 있으니 운영자에게 확인
403	<code>model_denied</code>	<code>model is not allowed by auth policy</code>	그 모델은 본인 키·팀에 허용되지 않음. 허용된 모델로 바꾸거나 운영자에게 정책 변경 요청
403	<code>provider_denied</code>	<code>provider is not allowed by auth policy</code>	<code>openai/...</code> 처럼 provider 를 명시했다면 허용된 provider 인지 확인
402	<code>budget_denied</code>	<code>estimated cost exceeds key budget limit</code>	이 한 건의 예상 비용이 키 예산을 넘음. 콘텐츠를 줄이거나 운영자에게 예산 조정 요청
402	<code>cost_threshold_exceeded</code>	<code>...resend with header 'X-Cost-Approve: 1' to proceed</code>	비싼 호출 승인 가드. 의도한 호출이면 헤더 <code>X-Cost-Approve: 1</code> 을 붙여 다시 보냄
429	<code>quota_error</code>	<code>quota exceeded: krw_limit_exceeded</code> 등	일·월 한도 도달. <code>Retry-After</code> 초 뒤 다음 기간에 풀림. 늘려야 하면 운영자에게 (4.3)
503	<code>kill_switch_active</code>	—	운영자가 긴급 정지를 켜 상태. 공지를 확인하고 기다림
502	<code>provider_unavailable</code>	<code>provider is unavailable</code>	업스트림 장애. 잠시 뒤 재시도. 계속되면 운영자에게

5.2 콘솔에서 막힐 때

- 로그인 뒤 메뉴가 거의 없다 — 역할이 `viewer` 나 팀 범위 계정이면 정상입니다. 필요한 화면은 운영자에게 역할을 요청하세요. SSO 로 들어왔는데 갑자기 줄었다면 운영자 가이드의 "SSO 로그인 후 메뉴가 거의 사라짐" 항목에 해당합니다.
- 화면에 "데이터를 불러오지 못했습니다" 와 요청 ID 가 뜬다 — 그 요청 ID 를 운영자에게 전달하세요. 서버 로그에서 같은 ID 로 찾을 수 있습니다.
- 기존 화면에서 열기 로 예전 콘솔이 열린다 — 해당 화면이 아직 새 콘솔로 옮겨지지 않았거나 롤아웃 대상이 아닌 것입니다. 기능은 같습니다.
- 세션이 만료되면 로그인 화면으로 돌아갑니다. 로그인하면 원래 보던 화면으로 되 돌아옵니다.

5.3 자주 묻는 질문

Q. 평소 쓰던 OpenAI 키를 그대로 써도 되나요? A. 아니요. `pcg_...` 형태의 proxy key 만 인증됩니다. OpenAI 키는 게이트웨이가 보관합니다.

Q. 응답이 갑자기 한국어로만 옵니까? A. 게이트웨이는 응답을 절대 수정하지 않습니다. 모델이 한국어로 답하는 것입니다.

Q. stream 응답이 끊기거나 늦습니다. A. 게이트웨이는 SSE 청크를 즉시 flush 합니다. 늦으면 네트워크 또는 upstream 자체의 문제입니다. `/health` 와 `/ready` 가 200 인지 확인하세요.

Q. trace_id 를 알면 어디서 볼 수 있나요? A. 게이트웨이가 모든 호출에 `X-Request-ID` 응답 헤더를 붙입니다. 그 값을 콘솔의 요청 탐색기 화면 "요청 ID" 필터에서 그대로 붙여넣으면 단건을 찾을 수 있습니다.

Q. 이전에 쓰던 키를 분실했어요. A. 한 번만 표시되므로 다시 볼 수 없습니다. 운영자에게 비활성화 + 새 키 발급을 요청하세요. 이전 키로 쌓인 통계는 그대로 보존됩니다.

Q. 사용량 알림을 받고 싶어요. A. 운영자에게 알림 규칙 추가를 요청할 수 있습니다 (지표 `requests/errors/kwr/tokens/latency_p95_ms/first_chunk_p95_ms/llm_eval_failures/llm_eval_failure_rate`, 윈도우 N초, 임계값, Slack 웹훅).

Q. 사용자별 이력이 전부 passthrough 나 anonymous 로 묶여요. A. 게이트웨이는 키의 해시만 저장하므로, 등록된 proxy key 로 호출해야 그 사용자로 정확히 집계됩니다. 운영자에게 사용자별 키 발급 (`PROXY_API_KEYS` 또는 콘솔 **사용자와 팀 → API 키**)을 요청하세요. 등록 없이 사용자별로 다른 키를 보내는 경우에도 게이트웨이가 키 지문으로 `ext_...` 사용자를 자동 분리합니다(같은 키=같은 사용자). 이때 `X-Vibe-User` (표시 이름)·`X-Vibe-Team` (팀) 헤더를 함께 보내면 사용자/팀 화면에 이름·팀이 표시됩니다. 모두가 같은 키(예: 공용 upstream 키)를 쓰면 한 사용자로 합쳐 지니, 분리가 필요하면 사용자마다 다른 키를 쓰세요.

6. 용어

용어	뜻
proxy key (pcg_ ...)	게이트웨이가 발급하는 개인·팀 키. 업스트림 vendor 키 대신 이것만 씁니다
provider / 공급자	게이트웨이 뒤의 LLM 서비스(openai, anthropic, 사내 vLLM 등). 모델명 앞에 openai/ 처럼 붙여 고를 수 있습니다
요청 ID (X-Request-ID)	호출 한 건의 식별자. 콘솔 검색과 운영자 문의에 씁니다
세션 (X-Vibe-Session)	에이전트가 이어서 보내는 여러 호출을 한 묶음으로 보는 단위
쿼터 / 예산	키·팀·IP 별 일·월 토큰 또는 KRW 한도. 넘으면 429
비용 가드	한 건의 예상 비용이 임계값을 넘으면 승인 헤더를 요구하는 장치
마스킹 ([REDACTED_...])	프롬프트·응답에서 비밀·개인정보를 라벨로 바꿔 저장하는 것
기존 화면 / 미리보기	예전 /admin 콘솔 화면 / 새 /app 콘솔로 옮겨진 화면
Kill Switch	운영자가 모든 호출을 즉시 막는 긴급 정지