

하나의 워크스페이스에서, 모든 작업을

코드 / 디자인 / 문서

작업 폴더와 모델을 연결하고, 코드를 실행하고, 실제 화면을 검증하고, 문서와 디자인 산출물까지 만드는 실전 사용 가이드입니다.

01 프로젝트 연결

개발 또는 문서/디자인 방식을 고르고 작업 폴더를 연결합니다.

02 모델 연결

로컬 CLI 또는 BYOK 공급자를 선택합니다. 비밀값은 프롬프트에 넣지 않습니다.

03 실행과 검증

빌드·서버·브라우저·Git 변경사항을 한 프로젝트 문맥에서 확인합니다.

04 산출물 완성

화면/인터페이스 명세, 슬라이드, 프로토타입을 미리보고 내보냅니다.

VERSION

2026.08 · Windows Desktop

처음 5 분: 프로젝트 하나를 끝까지 연결하기

아래 순서만 따르면 개발과 문서 제작 모두 같은 프로젝트 기록 안에서 시작할 수 있습니다.

단계	메뉴	해야 할 일
1	작업 방식	홈에서 '소프트웨어 개발' 또는 '문서/디자인 제작'을 선택합니다.
2	작업 폴더	개발은 실제 저장소 루트, 코드 기반 명세는 분석 대상 저장소를 선택합니다.
3	모델	설정에서 설치된 CLI 상태를 확인하거나 BYOK 공급자를 연결합니다.
4	첫 요청	개발은 실행·수정·검증을, 문서는 산출물 종류와 목적을 요청합니다.
5	검토	변경사항·브라우저·미리보기에서 결과를 확인한 뒤 내보내거나 커밋합니다.

안전한 기본값

- DB 는 읽기 전용으로 시작합니다. 쓰기가 필요하면 '쓰기마다 승인' 또는 '항상 쓰기 허용'을 선택합니다. 두 쓰기 모드도 구조화된 INSERT·UPDATE·DELETE 만 허용하며 DDL 과 임의 SQL 은 차단합니다.
- 브라우저 자동화는 탭별 승인 후 실행합니다. 외부 로그인과 민감 데이터는 사용자 확인이 우선입니다.
- 모델 표시 목록과 실제 사용 가능 여부는 CLI 버전·계정·공급자 권한에 따라 달라질 수 있습니다.
- 코드 기반 인터페이스 명세는 실제 코드베이스 폴더를 먼저 연결합니다. 화면 명세는 업로드하거나 캡처한 화면을 기준으로 작성합니다.

가이드

앱의 ? 메뉴에서 제품 가이드를 다시 열 수 있습니다. 첫 사용 시에는 현재 작업 방식에 맞는 화면 안내가 자동으로 시작됩니다.

작업 방식을 먼저 고르세요

하나의 프로젝트 안에서 작업 기록은 유지하되, 시작 화면과 기본 대화 모드는 목적에 맞게 달라집니다.

01 방식 선택

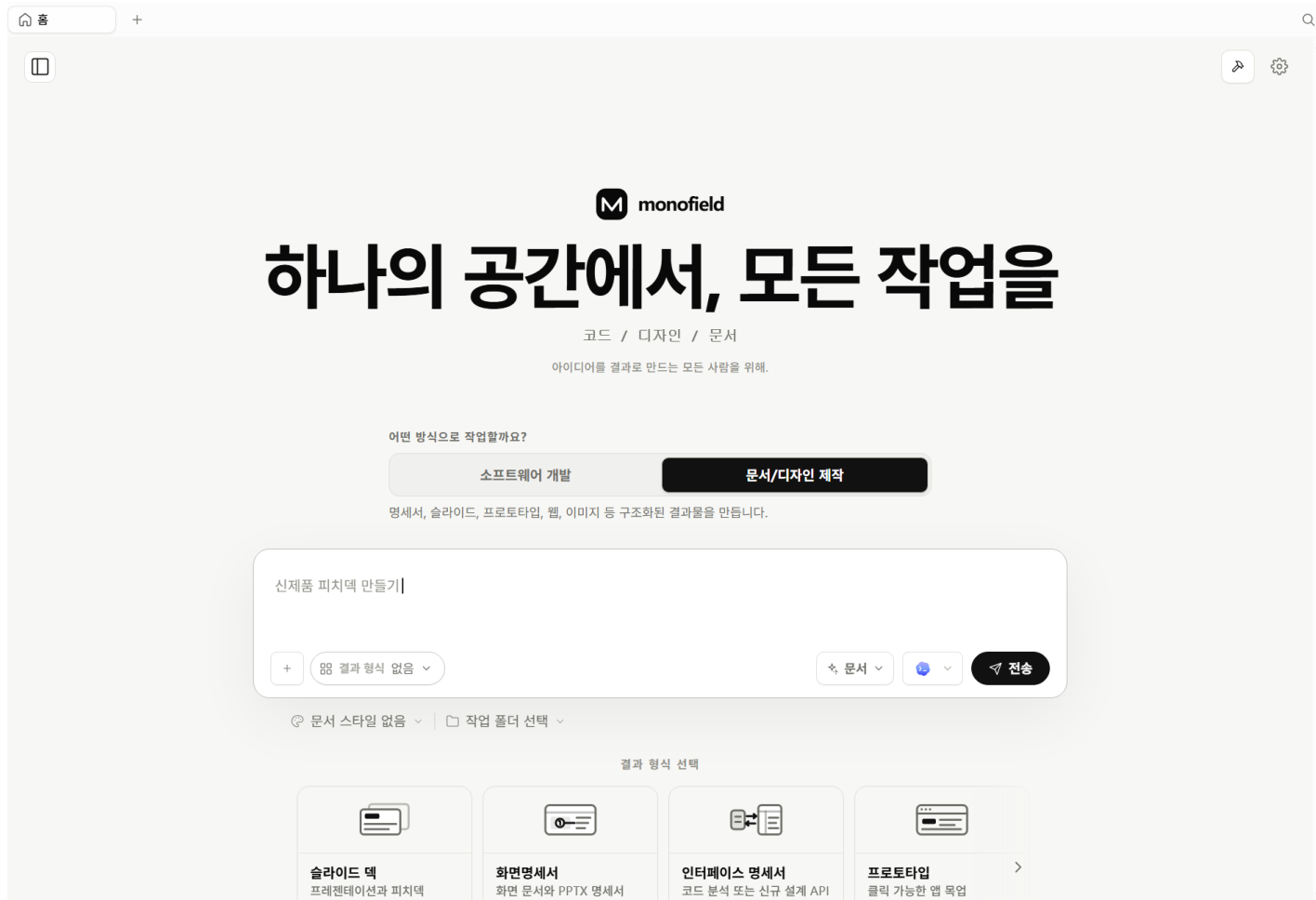
개발 또는 문서/디자인

02 폴더 연결

실제 작업 대상 폴더

03 대화 시작

질문 또는 문서 모드



실제 MonoField Desktop 화면 · 홈의 소프트웨어 개발 / 문서·디자인 선택

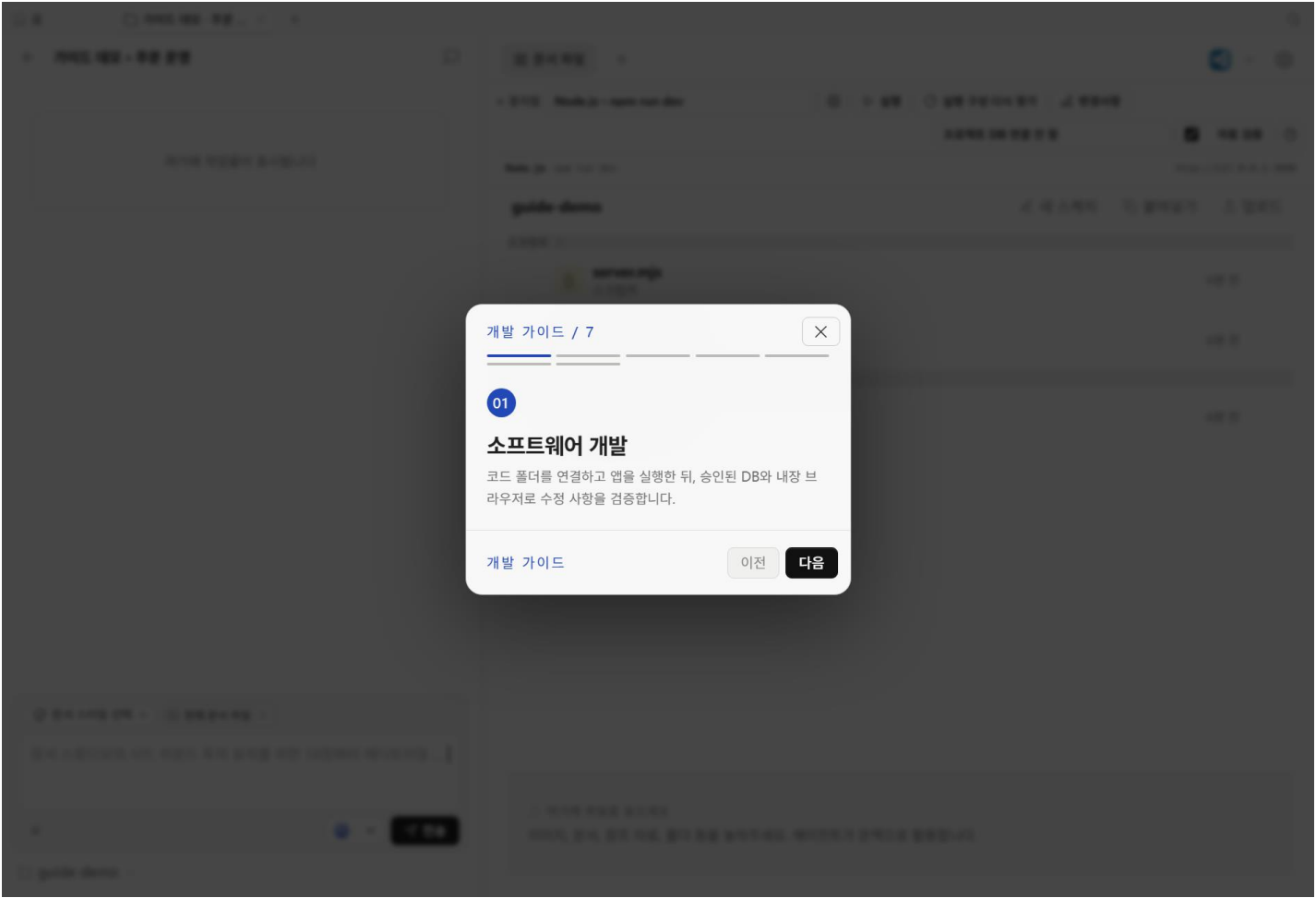
원칙

개발 프로젝트에서 기술 문서를 만들 때는 문서 모드로, 문서 프로젝트에서 설명·검토만 할 때는 질문 모드로 전환할 수 있습니다.

처음에는 화면이 직접 안내합니다

첫 사용 가이드는 나머지 화면을 어둡게 하고 지금 눌러야 할 요소를 강조합니다. 창 크기가 바뀌어도 대상 위치를 다시 계산합니다.

01 다음 강조된 요소 확인	02 직접 실행 안내에 따라 버튼 사용	03 다시 보기 ? → 제품 가이드
--------------------	--------------------------	------------------------



실제 MonoField Desktop 화면 · 개발 가이드 1/7 단계

팁	개발 가이드 다음에는 실행 구성·빌드·DB·변경사항·브라우저 검증 흐름을, 문서/디자인 가이드에서는 파일·스타일·미리보기·내보내기를 안내합니다.
---	--

작업 폴더가 개발의 기준입니다

선택한 폴더를 CLI 와 내장 터미널의 기본 작업 루트로 사용합니다. MonoField 가 실행 구성을 탐색하고 파일·Git·서버 로그·브라우저·DB 문맥을 같은 프로젝트에 연결합니다.

01 폴더 선택

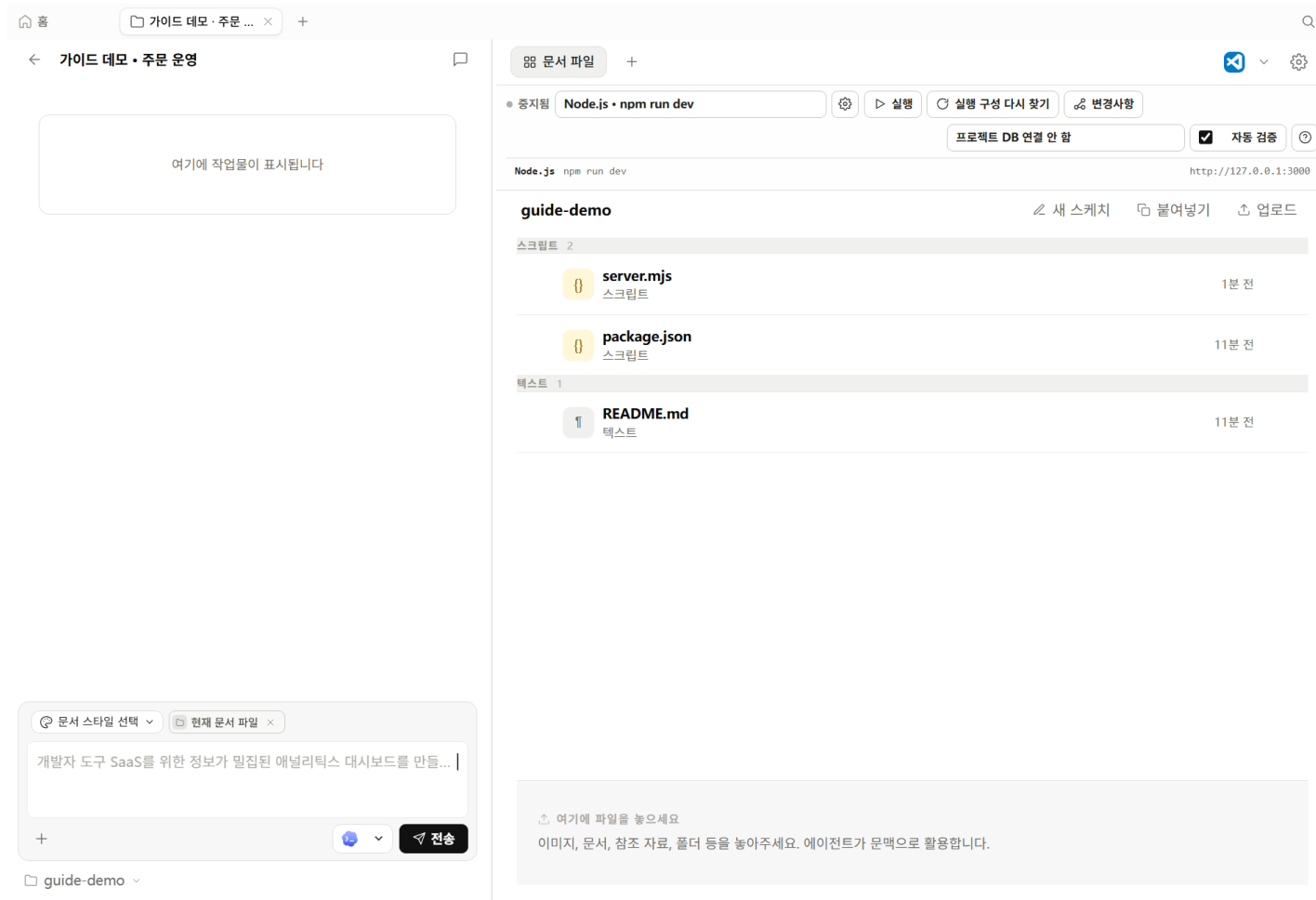
저장소 또는 워크스페이스

02 활성 프로젝트

여러 서버 중 하나 선택

03 터미널/대화

수정 → 테스트 → 화면 검증



실제 MonoField Desktop 화면 · 안전한 Node 데모 프로젝트와 감지된 실행 구성

확인

상위 워크스페이스를 선택하면 발견된 하위 프로젝트 중 활성 대상을 고릅니다. 실행·Git 감시는 활성 프로젝트에만 붙어 성능 부담을 줄입니다.

실행 전에 프로필과 인자를 확인하세요

감지된 기본 명령을 그대로 실행하거나 애플리케이션 인자.환경 변수를 프로젝트별 오버라이드로 저장할 수 있습니다.

01 구성 선택

Vite·Next·Maven·Gradle·Python·Go 등

02 프로필

예: --spring.profiles.active=local

03 실행/중지

상태와 로그 확인

여기에 작업물이 표시됩니다

실행 설정 편집

예: Spring Boot는 local/dev/prod 프로필과 --server.port=9081 같은 인자를 여기서 정합니다. 프로젝트의 application 파일을 수정하지 않고 이번 실행에만 적용됩니다.

추가 애플리케이션 인자

--debug --feature=orders

지우기 취소 저장

Node.js npm run dev http://127.0.0.1:3000

guide-demo 새 스케치 불러넣기 업로드

스크립트 2

server.mjs 스크립트 1분 전

package.json 스크립트 11분 전

텍스트 1

README.md 텍스트 11분 전

문서 스타일 선택 현재 문서 파일

기후 변화 관련 비영리 단체의 인터랙티브 연례 보고서를 제작해 주세요...

전송

guide-demo

여기에 파일을 놓으세요

이미지, 문서, 참조 자료, 폴더 등을 놓아주세요. 에이전트가 문맥으로 활용합니다.

실제 MonoField Desktop 화면 · 추가 애플리케이션 인자 편집

주의

실행 버튼 옆 구성명을 확인하세요. 다른 서버가 열리면 활성 프로젝트, 포트 인자, 환경 변수를 먼저 확인합니다.

변경 전후를 화면에서 검토합니다

변경사항에서는 작업 트리·스테이징·브랜치/커밋 비교를 전환하고 파일별 Before/After 를 가로 스크롤로 확인합니다.

01 기준 선택

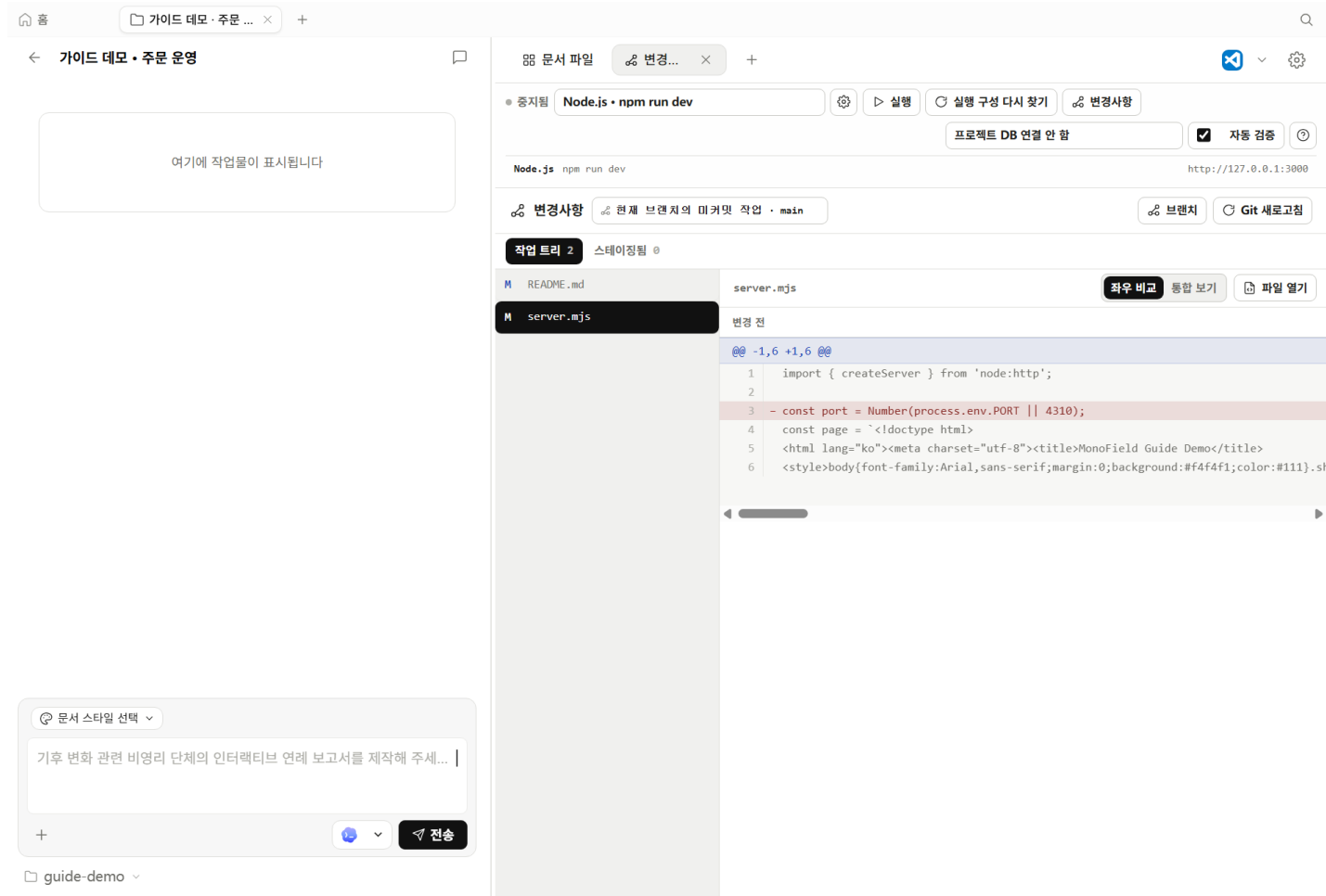
현재 작업 또는 브랜치/커밋

02 파일 선택

추가·수정·삭제 확인

03 검토

Before ↔ After 가로 스크롤



실제 MonoField Desktop 화면 · 실제 Git diff 와 Before/After 영역

구분

현재 작업은 마지막 커밋 이후의 로컬 변경입니다. 브랜치/커밋 비교는 선택한 Git 기준점과 현재 상태의 차이입니다.

서버를 실행하고 실제 동작을 검증합니다

실행 상태·로그·URL 을 확인한 뒤 내장 브라우저로 엽니다. 자동화는 탭별 권한을 승인해야 클릭·입력·스크롤·업로드·드래그를 수행합니다.

01 실행

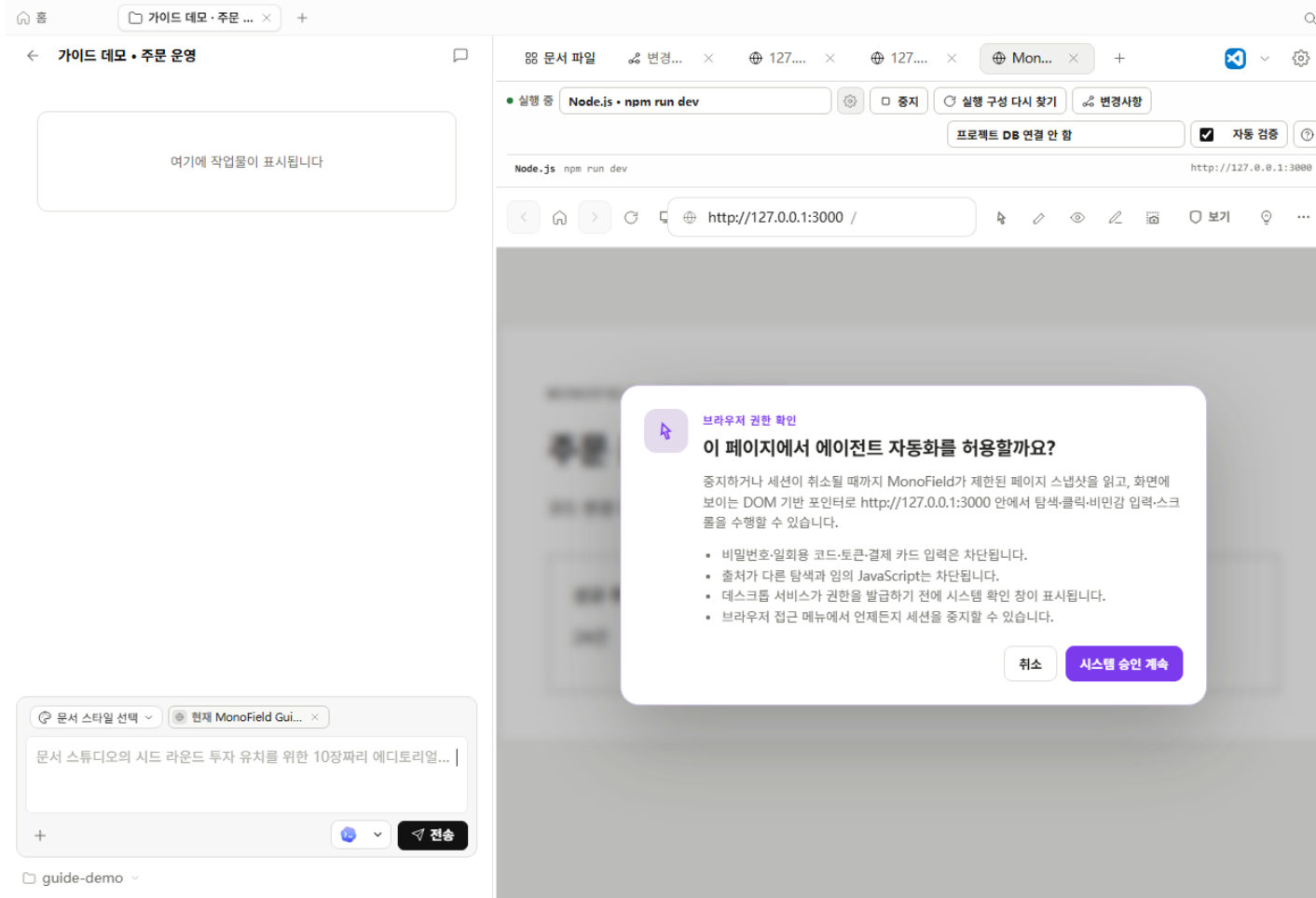
PID·포트·로그 표시

02 브라우저

로컬/원격 URL 열기

03 권한 승인

자동화 대상 탭을 명시



실제 MonoField Desktop 화면 · 실행 중인 서버와 브라우저 자동화 승인

보안

승인하지 않은 탭은 조작하지 않습니다. 자동화 권한은 사용자 프로필에 저장할 수 있지만, 민감한 동작은 요청 범위 안에서만 수행해야 합니다.

보이는 문제를 그대로 수정 요청에 답하세요

브라우저 화면에서 표시·화면에 그리기·검사·조정을 여러 개 쌓은 뒤 한 번에 수정 요청으로 보낼 수 있습니다.

01 표시

DOM 요소를 선택

02 그리기

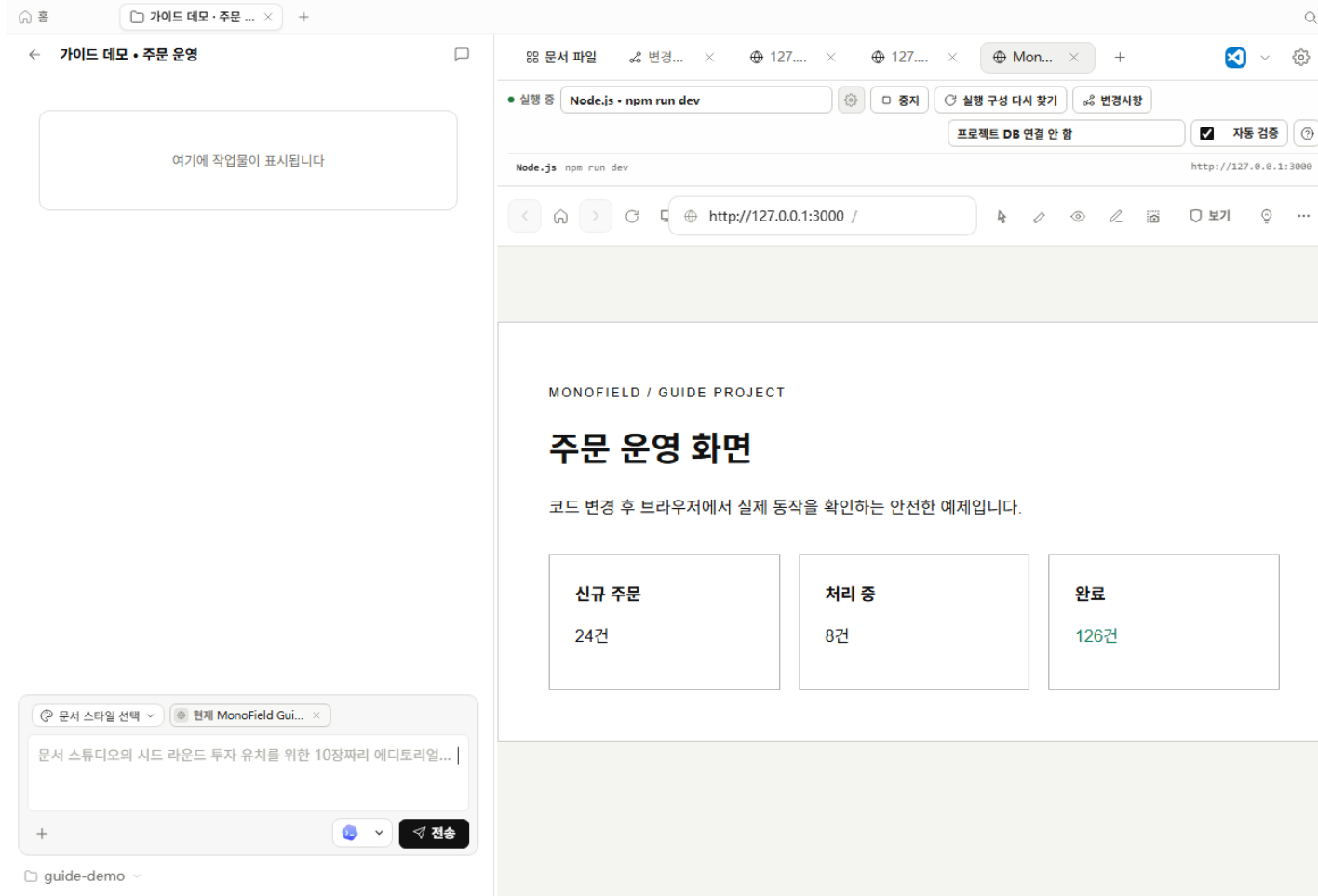
영역과 메모를 캡처

03 검사/조정

구조·스타일 변경안 추가

04 일괄 요청

모든 항목을 CLI 에 전달



실제 MonoField Desktop 화면 · 실행 중인 주문 화면의 브라우저 검증

자동 검증

활성화하면 수정 후 빌드와 승인된 브라우저의 실제 화면·동작 확인을 다음 검증 단계로 사용합니다. 단순 새로고침 옵션이 아닙니다.

설치된 CLI 와 모델을 확인하세요

MonoField 는 CLI 의 도움말·모델 명령을 읽어 현재 설치 상태와 사용 가능한 모델을 표시합니다. 알려진 모델 목록은 오프라인 폴백으로만 사용합니다.

01 설치 확인

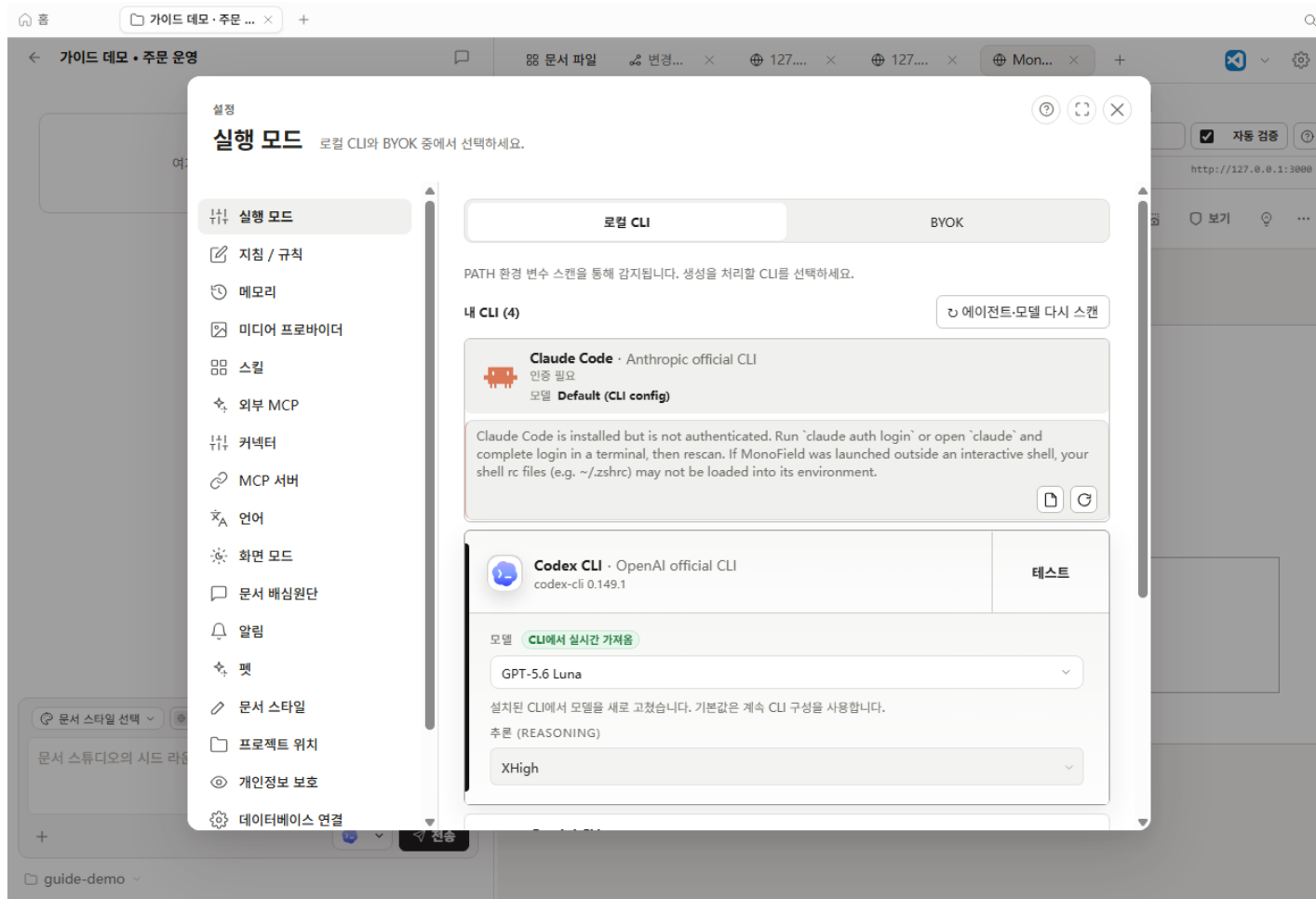
Codex·Claude·Gemini·OpenCode·Copilot 등

02 인증

각 CLI 의 공식 로그인 절차

03 모델 새로고침

설치 버전에서 다시 발견



실제 MonoField Desktop 화면 · 실행 모드에서 확인한 로컬 CLI 상태

가용성

목록에 보여도 계정 권한이나 지역·조직 정책에 따라 호출이 거절될 수 있습니다. 새 모델은 CLI 를 업데이트한 뒤 모델 목록을 새로 고치세요.

API 키와 OpenAI 호환 엔드포인트를 연결합니다

Anthropic·OpenAI·Google·OpenRouter·OpenAI-compatible endpoint 를 공급자별로 구성합니다. 로컬 Ollama 는 API 키 없이 모델을 발견할 수 있습니다.

01 공급자

프리셋 또는 사용자 지정

02 엔드포인트

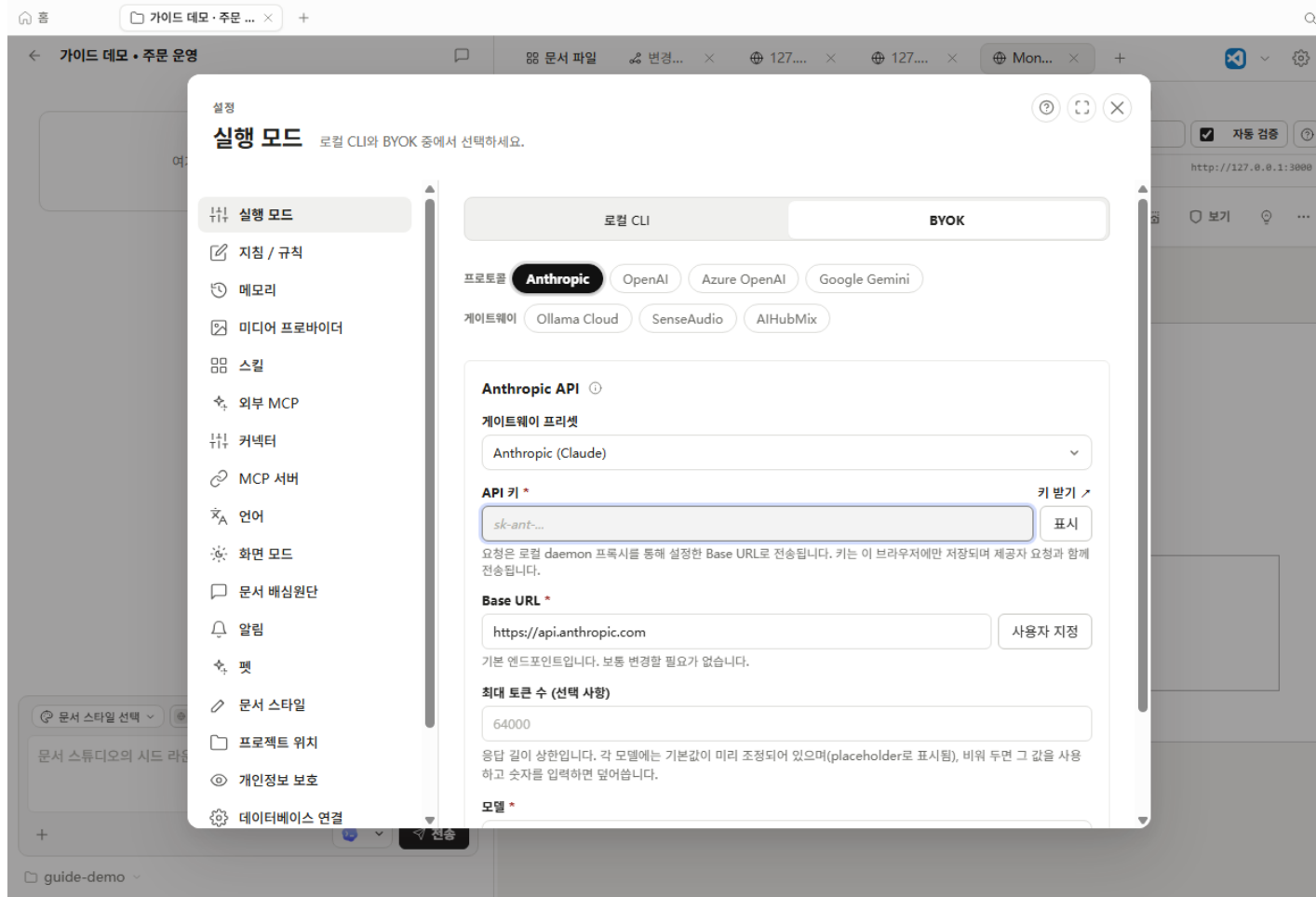
공식/게이트웨이/사내 URL

03 연결 테스트

모델 조회와 짧은 요청

04 저장

Desktop 암호화 저장소



실제 MonoField Desktop 화면 · BYOK 공급자와 프로토콜 설정

비밀값

API 키와 DB 자격 증명은 프롬프트·프로젝트 파일에 저장하지 않습니다. 프로젝트에는 연결 ID 만 남기고 실제 비밀값은 Desktop 보안 저장소에서 해석합니다.

문서와 디자인도 같은 프로젝트 문맥에서 만듭니다

문서/디자인 제작에서는 빈 캔버스, 파일, 문서 스타일, 템플릿을 조합해 명세서·슬라이드·프로토타입·반응형 웹을 제작합니다.

01 자료

파일·용어사전·브랜드 시스템

02 형식

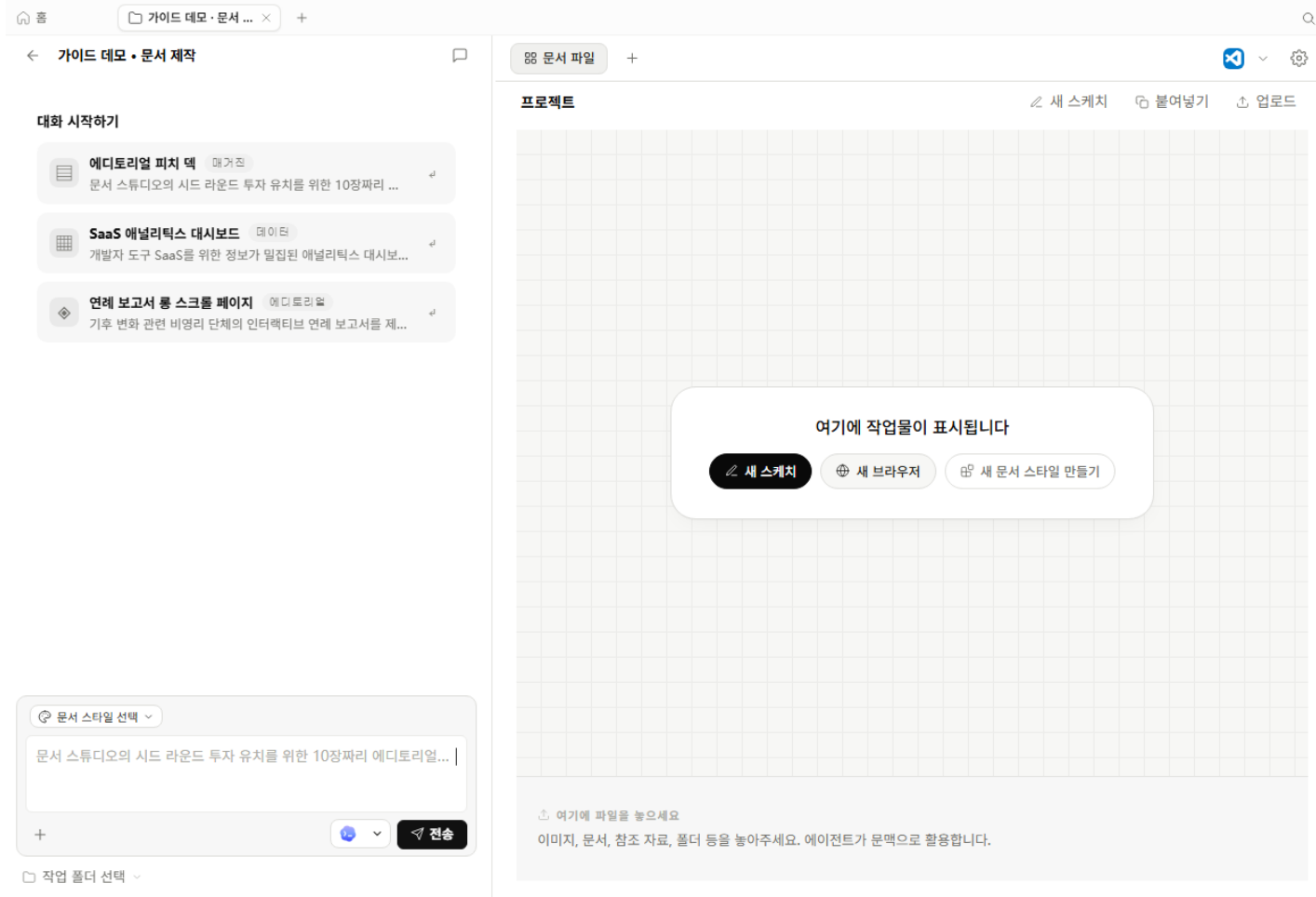
명세서·슬라이드·웹·와이어프레임

03 AI 초안

빈 필드는 합리적 초안 생성

04 검토/내보내기

미리보기 → 편집 → export



실제 MonoField Desktop 화면 · 문서/디자인 작업 공간의 시작 카드

명세서

코드 기반 인터페이스 명세는 실제 코드 폴더 검증 → 옵션 품 → 스캔 → XLSX 흐름입니다. 코드 없는 인터페이스는 최소 정보로 AI 초안을 만든 뒤 제안을 검토·확정해야 내보낼 수 있습니다. 화면 명세는 업로드/캡처 화면에 마커·Description·Check Point 를 편집하고 HTML 미리보기 후 PPTX 로 내보냅니다.

어떤 요청을 어디에서 시작할까요?

요청의 목적에 맞춰 작업 방식과 대화 모드를 바꾸면 불필요한 코드 스캔과 질문을 줄일 수 있습니다.

하려는 일	작업 방식	대화 모드	권장 요청
코드 수정·테스트	소프트웨어 개발	질문	‘주문 생성 오류를 고치고 테스트와 브라우저 검증까지 해줘.’
기술 명세서	소프트웨어 개발	문서	‘현재 코드를 기준으로 인증 모듈 기술 명세서를 만들어줘.’
새 문서·디자인	문서/디자인 제작	문서	‘업로드한 주문 화면을 마커·Description·Check Point 가 포함된 화면 명세서로 만들어줘.’
문서 검토·설명	문서/디자인 제작	질문	‘이 명세서의 누락과 모순만 검토해줘.’
코드 기반 인터페이스 명세	문서/디자인 제작	문서	대상 코드베이스 폴더 연결 → 옵션 선택 → 스캔 → xlsx 미리보기/내보내기
코드 없는 인터페이스 명세	문서/디자인 제작	문서	Method·Path·ID·인증을 주고, 요청/응답 필드는 비워 AI 초안을 요청할 수 있습니다.

출시 버전 사용 전 점검

- 설정 → 실행 모드에서 원하는 CLI 가 ‘사용 가능’인지 확인합니다.
- BYOK 는 연결 테스트 후 모델 목록을 새로 고칩니다. 키가 없는 로컬 Ollama 도 사용할 수 있습니다.
- 업데이트 알림은 GitHub Release 의 최신 설치 파일을 기준으로 동작합니다.
- Windows 가 설치 파일을 경고하면 GitHub Release 의 해시와 게시자를 확인합니다. 정식 코드 서명 전에는 SmartScreen 경고가 남을 수 있습니다.

도움말

문제가 반복되면 실행 중인 프로젝트, 선택한 실행 구성, CLI 버전, 모델명, 브라우저 승인 상태를 함께 확인하세요.

여러 서버를 한 워크스페이스에서 관리합니다

상위 폴더를 열면 MonoField 가 실행 가능한 하위 프로젝트를 발견합니다. 활성 프로젝트를 바꿔도 각 서버의 PID·URL·로그·실행 구성이 섞이지 않습니다.

01 워크스페이스 여러 저장소가 들어 있는 상위 폴더 선택	02 활성 프로젝트 여러 프로젝트 중 현재 작업 대상 선택	03 독립 실행 프로젝트별 구성·프로필·포트로 실행	04 상태 복원 다시 선택하면 기존 로그와 실행 상태 확인
---	--	--	---

표시	의미	사용자가 할 일
초록 · 실행 중	해당 프로젝트 프로세스가 살아 있고 URL 또는 로그가 연결됨	로그를 보거나 브라우저에서 검증
파랑 · 불러오는 중	실행 구성·Git 상태·서버 상태를 비동기로 갱신 중	전환을 반복하지 말고 완료 표시 확인
회색 · 중지됨	구성은 발견됐지만 현재 실행 프로세스가 없음	프로필과 포트를 확인한 뒤 실행
빨강 · 실패	포트 충돌, 누락된 런타임, 명령 실패 등이 로그에 기록됨	실행 로그의 전체 오류를 열어 원인 수정

성능	전체 트리를 매번 재귀 스캔하지 않습니다. 알려진 프로젝트 표식만 제한적으로 찾고, 무거운 폴더는 제외하며, Git 감시와 상세 로딩은 활성 프로젝트에 집중합니다.
----	---

DB 는 연결보다 권한이 먼저입니다

프로젝트에는 연결 ID 만 남고 실제 자격 증명은 Desktop 암호화 저장소에 보관됩니다. 멀티 프로젝트에서는 활성 모듈마다 연결과 정책을 따로 선택합니다.

01 읽기 스키마·키·허용된 샘플 조회	02 승인형 쓰기 INSERT·UPDATE·DELETE 마다 사용자 승인	03 항상 허용 선택한 DB 와 프로젝트 범위에서 구조화된 쓰기	04 감사 대상·정책·결과를 로컬 기록으로 확인
---------------------------------	--	---	--------------------------------------

정책	허용 범위	권장 상황
읽기 전용	스키마와 제한된 SELECT	운영 분석, 명세서, 처음 연결한 DB
쓰기마다 승인	구조화된 DML 을 요청별 승인	개발·테스트 데이터 수정
항상 쓰기 허용	동일 DB identity 와 활성 프로젝트에 한해 구조화된 DML	격리된 개인 개발 DB 의 반복 테스트

보안	같은 이름으로 다른 주소의 DB 를 다시 저장해도 기존 쓰기 권한은 승계되지 않습니다. DDL 과 임의 SQL 은 정책과 무관하게 차단되며, 운영 DB 라고 자동 추정해 별도 권한을 부여하지 않습니다.
-----------	--

코드가 있어도, 없어도 인터페이스 명세를 만듭니다

두 흐름은 입력 근거가 다릅니다. 코드 기반 흐름은 실제 소스를 먼저 검증하고, 신규 설계 흐름은 최소 필드만 받은 뒤 비어 있는 요청·응답을 AI 초안으로 보완합니다.

구분	필수 입력	결정적 흐름	결과
코드 기반	분석 가능한 코드베이스 폴더	폴더 검증 → 옵션 → 코드 스캔 → JSON	XLSX 미리보기·export
신규 설계	Method·Path·Interface ID	인증·필드·크기 → AI 초안 → 사용자 검토	기본/사내 템플릿 XLSX
설명 질문	질문 문장	코드·DB·품을 열지 않고 일반 설명	대화 답변

요청 예시

코드 없이 주문 생성 인터페이스 명세서를 만들어줘. POST /api/orders, IF-ORD-001, Bearer 인증. customerId 와 items 를 받고 orderId 와 status 를 응답해.

- 요청·응답 필드는 비워 둘 수 있습니다. 연결한 용어사전·템플릿·참고 문서를 우선 근거로 쓰고, 없으면 합리적 초안을 제안합니다.
- 최소·최대 크기는 처음 입력하거나 초안 이후 검토 단계에서 조정할 수 있습니다.
- 코드 기반 요청에서 작업 폴더가 없으면 옵션 품·코드 스캔·빈 명세 생성을 시작하지 않습니다.

실제 화면을 근거로 화면명세서를 만듭니다

업로드 이미지, 브라우저 캡처, 코드와 용어사전을 함께 연결할 수 있습니다. 화면마다 Description 과 Check Point 를 검토한 뒤 XLSX 또는 PPTX 로 내보냅니다.

01 수집	02 구조화	03 검토	04 납품
화면 업로드·브라우저 캡처·DOM 표시	화면 ID·설명·체크포인트·전환 정의	미리보기에서 표시·그리기·편집·댓글	검증 후 XLSX·PPTX export

기능	언제 쓰나	결과
표시	버튼·문구처럼 DOM 으로 식별 가능한 요소	선택자와 화면 위치가 수정 요청에 연결
화면에 그리기	레이아웃 간격·영역처럼 시각적으로 설명할 부분	영역 캡처와 메모가 하나의 검토 항목으로 저장
검사	구조·스타일·접근성 근거가 필요한 경우	DOM 정보와 현재 스타일을 함께 전달
조정	32px→28px 처럼 구체적인 변경안을 남길 때	다른 항목과 순번을 공유해 일괄 요청

파일 안전

화면 이미지 참조는 프로젝트 안의 일반 PNG·JPEG·WebP 만 허용하며, 실제 MIME 과 확장자를 확인하고 20MiB 를 넘는 파일과 심볼릭 링크 경로 탈출을 차단합니다.

토큰은 마지막 문장 길이가 아니라 전체 모델 문맥입니다

사용량 패널의 입력은 CLI 또는 모델 공급자가 이번 응답에 보고한 시스템 지침·도구 정의·대화 기록·선택된 프로젝트 문맥의 합계입니다. MonoField 가 별도로 같은 토큰을 한 번 더 청구한다는 뜻이 아닙니다.

항목	뜻	해석
새 입력	이번 호출에서 새로 처리한 입력	사용자 문장 외 시스템·도구·대화 문맥 포함
캐시 재사용	공급자가 이전 입력 prefix 를 재사용한 양	중복 전송처럼 보여도 공급자 캐시 지표이며 보통 새 입력보다 효율적
출력·추론	응답과 내부 추론에 보고된 양	런타임이 구분해 주지 않으면 측정 불가로 표시
비용	공급자가 직접 반환한 USD	CLI 가 금액을 주지 않으면 MonoField 가 임의 환산하지 않음

도구 차이

로컬 코드-agent CLI 는 파일·터미널·DB broker·브라우저 검증과 함께 사용합니다. 일반 BYOK endpoint 는 대화와 문서 생성에 적합하며, 공급자 API 만 연결했다고 로컬 개발 도구 권한이 자동으로 생기지는 않습니다.

- 문서 모드 의 긴 지침은 실제 문서 생성·검증 작업에서만 지연 주입합니다.
- 선택한 CLI 의 준비 상태를 먼저 표시하고, 다른 에이전트 전체 스캔은 뒤에서 진행합니다.
- 모델 목록은 설치된 CLI 의 실시간 결과를 우선하고 알려진 목록은 오프라인 폴백으로만 사용합니다.

막혔을 때는 현재 프로젝트의 상태부터 확인하세요

오류를 숨기고 다른 프로젝트를 임의로 실행하지 않습니다. 실행 로그와 현재 선택값을 기준으로 한 단계씩 복구합니다.

증상	확인	조치
Port already in use	실행 로그의 전체 포트 충돌 메시지	포트 인자를 바꾸거나 해당 포트 프로세스를 명시적으로 중지
다른 서버가 열림	활성 프로젝트·절대 작업 경로·프로필·URL	올바른 모듈 선택 후 실행 구성 다시 찾기
UNC 폴더 쓰기 실패	Desktop 권한과 프로젝트 root	프로젝트 파일 API 를 사용하거나 로컬 작업 폴더 선택
모델 호출 실패	CLI 로그인·모델 권한·공급자 상태	설정에서 다시 스캔/테스트 후 재시도, 반복되면 이슈 또는 메일
가이드가 안 보임	? 도움말과 각 화면의 가이드 버튼	제품·개발·문서/디자인·설정 가이드를 수동으로 다시 열기
업데이트 발견	앱의 새 버전 알림과 GitHub Release	해시를 확인하고 최신 설치 파일 다운로드

문의 시 포함

MonoField 버전, 활성 프로젝트 경로, 실행 구성명, CLI 와 모델명, 실패 시각, 실행 로그 또는 진단 export 를 함께 보내면 재현이 빨라집니다.

출시 전, 이 순서로 한 번만 확인하세요

개발과 문서/디자인 모두 같은 원칙을 사용합니다. 입력 근거를 명시하고, 실행 상태를 확인하고, 결과를 눈으로 검토한 뒤 내보냅니다.

단계	개발	문서/디자인
1. 프로젝트	작업 폴더와 활성 모듈 확인	작업 폴더·참고 파일·용어사전 확인
2. 엔진	CLI 로그인·모델·추론 설정 확인	CLI 또는 BYOK 연결 테스트
3. 권한	DB 정책과 브라우저 탭 승인 확인	코드/브라우저/DB 근거가 필요한지 확인
4. 실행	프로필·인자·포트·로그 확인	템플릿·스타일·필수 필드 확인
5. 검토	diff·테스트·실제 화면 동작 확인	미리보기·표시·편집·검증 확인
6. 완료	커밋·배포·회귀 결과 기록	XLSX·PPTX·PDF·HTML export 확인

다시 보기

앱의 ? 메뉴에서는 제품 가이드를, 개발 바의 ?에서는 개발 가이드를, 문서 파일 탭의 ?에서는 문서/디자인 가이드를 다시 열 수 있습니다. 설정과 좌측 기능도 처음 열 때 각 기능의 역할을 안내합니다.