

Codex AGENTS.md

운영 핸드북

지침 계층, 지속성 검증, 외부 모델 게이트, 자동 복구

파일을 쓰는 데서 끝내지 않고 새 세션에서도 같은 경계를 증명하는 실전 운영 매뉴얼

v1.0 · 2026

이 책의 목적

AI 코딩 에이전트에게 긴 지침을 주는 것과 그 지침이 실제로 적용되는 것은 다른 문제입니다. 파일을 잘 작성해도 잘못된 위치에 있거나, 더 가까운 지침에 덮이거나, 로딩 한도에서 잘리면 에이전트는 규칙의 일부만 보고 작업합니다.

이 책은 Codex의 AGENTS.md를 팀 운영 장치로 만드는 방법을 다룹니다. 목표는 멋진 프롬프트가 아니라 새 세션에서도 같은 경계와 검증 절차가 재현되는 시스템입니다.

이 가이드는 특정 도구의 성공이나 수익을 보장하지 않습니다. 계정 생성, 약관 동의, 결제, KYC, 비밀번호 처리 같은 명의 행위는 반드시 권한 있는 사람이 수행해야 합니다.

이 책에서 만드는 것

- 전역, 프로젝트, 하위 디렉터리 지침의 역할 분리
- 충돌과 덮어쓰기를 찾는 로딩 경로 점검
- 긴 지침이 잘리는 문제를 사전에 탐지하는 용량 게이트
- 새 프로세스가 규칙을 실제로 읽었는지 증명하는 지속성 테스트
- 외부 모델을 최신 버전으로만 호출하는 실패 폐쇄형 오케스트레이션
- 백그라운드 작업의 건강 상태, 라우팅, 자동 복구 설계
- 작업 종료 전에 증거와 외부 차단 요소를 분리하는 운영 체크리스트

1. 지침은 문서가 아니라 실행 경계다

좋은 지침은 “최선을 다하라”가 아니라 무엇을 언제 확인하고, 어떤 조건에서 멈추며, 무엇을 증거로 남길지 정의합니다. 문장 하나가 모호하면 에이전트는 합리적으로 보이는 다른 행동을 선택할 수 있습니다.

약한 규칙과 강한 규칙

약한 규칙은 선호를 말합니다.

약한 규칙: 가능하면 최신 모델을 사용합니다.

강한 규칙은 선택 절차와 실패 결과를 고정합니다.

- 공식 릴리스와 현재 API 카탈로그에서 최신 GA 모델을 확인합니다.
- 로컬 래퍼가 정확한 모델 ID를 선택할 수 있어야 합니다.
- 확인되지 않으면 구버전 대체 없이 실패 폐쇄합니다.
- 모델, 근거 URL, 결과 상태를 비밀정보 없이 기록합니다.

두 번째 규칙은 에이전트가 기억이나 별칭에 의존할 공간을 줄입니다. 운영 지침에서 중요한 것은 문체가 아니라 판단 가능성의 범위입니다.

한 규칙에 들어가야 할 다섯 요소

- 트리거: 언제 적용되는가
- 입력: 무엇을 확인하는가
- 순서: 어떤 게이트를 먼저 통과하는가
- 실패 결과: 무엇을 하지 않는가
- 증거: 무엇을 남겨야 완료인가

2. 지침 계층을 역할별로 나눈다

Codex 작업에서는 보통 전역 지침, 저장소 지침, 하위 디렉터리 지침이 함께 존재합니다. 모든 내용을 전역 파일에 넣으면 프로젝트 고유 규칙이 다른 저장소까지 번지고, 모든 내용을 프로젝트 파일에 넣으면 반복되는 안전 경계가 누락됩니다.

전역 지침에 둘 것

- 비밀정보 출력 금지
- 파괴적 명령의 승인과 대상 확인
- 외부 모델 버전 확인과 실패 폐쇄 정책
- 작업 계획, 검증, 인수인계의 공통 게이트
- 전역 작업 기록과 지속성 검증 방법

프로젝트 지침에 둘 것

- 제품 목표와 비목표
- 기술 스택과 테스트 명령
- 배포 대상과 운영 환경
- 저장소의 디렉터리 경계
- 이 프로젝트만의 계정, 법적, 데이터 제한

하위 디렉터리 지침에 둘 것

- 특정 패키지의 코드 스타일
- 특정 언어 또는 도구의 검사 명령
- 산출물 위치와 파일 형식
- 해당 경로에서만 유효한 예외

가까운 지침이 더 구체적이어야 합니다. 더 가까운 파일이 전역 안전 규칙을 약화하지 못하도록 "프로젝트 규칙은 전역 게이트를 강화할 수 있지만 우회할 수 없다"는 문장을 전역에 둡니다.

3. 실제 로딩 경로를 증명한다

파일이 존재한다는 사실은 로딩을 증명하지 않습니다. 에이전트가 읽는 홈 디렉터리와 사용자가 편집한 홈 디렉터리가 다를 수 있고, 심볼릭 링크나 런타임 복사본이 오래된 파일을 가리킬 수도 있습니다.

점검 순서

1. 현재 프로세스의 CODEX_HOME을 확인합니다. 2. 그 홈 아래의 AGENTS.md가 실제 파일인지 링크인지 확인합니다. 3. 링크라면 최종 대상 경로를 확인합니다. 4. 전역 파일과 런타임 파일의 바이트가 같은지 비교합니다. 5. 비어 있지 않은 AGENTS.override.md가 있는지 찾습니다. 6. 현재 작업 디렉터리부터 루트까지 프로젝트 지침을 나열합니다.

macOS 점검 예시

```
printf '%s\n' "$CODEX_HOME" 로 홈을 확인하고, readlink "  
$CODEX_HOME/AGENTS.md" 로 최종 대상을 확인합니다. cmp -s로 전역 파일과 런타임 파일을  
비교한 뒤 비어 있지 않은 AGENTS.override.md를 찾습니다.
```

cmp가 성공해도 아직 끝이 아닙니다. 이것은 파일 동일성만 보여주며, 새 에이전트 프로세스가 내용을 주입받았다는 사실은 보여주지 않습니다.

4. 긴 지침의 잘림을 막는다

지침 파일이 커지면 로더의 문서 용량 제한에 걸릴 수 있습니다. 위험한 점은 앞부분은 정상적으로 보이는데 뒤쪽의 최신 모델 정책이나 금지 규칙만 사라질 수 있다는 것입니다.

용량 게이트

- 전역 지침 파일의 바이트 크기를 측정합니다.
- 프로젝트와 상위 경로에서 함께 주입되는 지침의 크기를 합산합니다.
- 설정의 `project_doc_max_bytes`가 전체 체인보다 충분히 큰지 확인합니다.
- 한도를 바꾼 뒤 새 프로세스로 다시 검증합니다.

```
macOS에서는 stat -f '%z' "$HOME/.codex/AGENTS.md"로 파일 크기를, rg '^project_doc_max_bytes' "$HOME/.codex/config.toml"로 로딩 한도를 확인합니다.
```

한도는 파일 하나의 크기와 같게 잡지 않습니다. 프로젝트 지침과 플러그인 주입 내용이 더해질 수 있으므로 여유를 둡니다. 한도를 무한히 키우는 것도 답이 아닙니다. 전역 지침은 공통 안전 경계에 집중하고, 프로젝트 세부사항은 가까운 파일로 내립니다.

잘림 사고의 신호

- 규칙의 앞부분만 반복해서 지켜집니다.
- 최근 추가한 뒤쪽 금지 항목만 무시됩니다.
- 같은 세션에서는 되지만 새 세션에서 사라집니다.
- 런타임 로그에 문서 또는 컨텍스트 축약 경고가 나타납니다.