



Cursor AI 코드베이스 인덱싱 한계와 최적화 전략

Cursor AI는 로컬 코드베이스를 인덱싱해서 대규모 프로젝트에서도 빠르게 맥락을 이해하고 작업을 도와주는 AI 코딩툴입니다. 하지만 인덱싱에는 구조적 한계가 있으며, 이를 이해하고 최적화하는 것이 중요합니다.

1. 인덱싱 가능한 최대 파일 수

- Cursor는 최대 약 100,000개의 파일까지만 인덱싱할 수 있습니다. 이를 초과하면 Too many files to upload 오류가 발생할 수 있습니다.
출처: [Reddit - Too many files to upload](#)
- .cursorignore 파일을 통해 인덱싱할 파일을 선택적으로 관리하면 이 한계를 효율적으로 우회할 수 있습니다.
출처: [Cursor 공식 문서 - Codebase Indexing](#)

2. 인덱싱 방식: 벡터 임베딩

- Cursor는 각 파일을 작은 코드 청크로 나눈 뒤, 이를 벡터화하여 벡터 데이터베이스에 저장합니다. 실제 코드가 저장되는 것이 아니라, AI가 의미를 이해할 수 있도록 벡터 임베딩만 저장됩니다.
출처: [Pixelstech - Windsurf indexing logic](#)
- 따라서 Cursor는 전체 코드를 한 번에 기억하진 않고, 벡터 검색을 통해 관련 청크만 찾아서 사용합니다.

3. 최적화 전략

- .cursorignore 사용: .gitignore처럼 작동하여 불필요한 폴더나 대용량 파일은 인덱싱에서 제외
출처: [Cursor 공식 문서 - Ignore Files](#)
- .cursorindexingignore 사용: 해당 파일은 인덱싱에서는 제외되지만 AI는 직접 참조 가능 (보조용).
출처: [Reddit - cursorignore vs cursorindexingignore](#)
- 파일 크기 관리: 가능한 한 500줄 이하로 유지하면 인덱싱 효율 향상
출처: [Pixelstech - Windsurf indexing logic](#)
- 명시적 참조 (@filename): 코드나 파일명을 명시하면 AI가 정확히 그 파일을 참조하도록 유도 가능

4. 보안 및 프라이버시

- .cursorignore는 사용자의 의도를 최대한 반영하려 하지만, 민감한 정보는 직접 관리가 필요합니다.
출처: [Medium - Refine에서 .cursorignore 사용](#)
- 벡터화된 데이터는 코드 원문이 아닌 요약 벡터이므로 기본적인 보안은 확보되지만, 코드 자체는 로컬에 존재합니다.
출처: [GitHub - getcursor 이슈](#)

✓ 요약

항목	내용
최대 인덱싱 수	약 100,000개 파일
인덱싱 방식	벡터 임베딩 기반
최적화	.cursorignore, .cursorindexingignore, 명시적 참조
보안	민감 정보는 제외 권장