

fixhub 도입 매뉴얼 (사이트 운영자용)

fixhub는 웹사이트 화면에서 쓴 개선 리포트를 Claude Code 또는 Codex가 받아 처리하는 서비스입니다.

처음 붙이는 데 필요한 것은 아래 5단계가 전부입니다. 예시는 `man.bebo.kr` (NabiMan)입니다.

필요한 것	설명
fixhub 계정	콘솔 로그인용. https://fix.bebo.kr/console/ 의 회원가입 신청 후 운영자 승인, 또는 이미 쓰는 조직의 초대 링크
사이트 HTML을 고칠 권한	<code></body></code> 앞에 스크립트 한 줄을 넣습니다
Claude Code 또는 Codex가 설치된 서버	사이트 소스 폴더에서 실행할 수 있어야 합니다. Claude Code는 자동 감시, Codex는 <code>\$fixhub</code> 스킬로 요청 시 처리합니다

1단계. 계정 만들기 (처음 한 번)

- <https://fix.bebo.kr/console/> → 회원가입 신청을 누릅니다.
- 이메일, 이름, 비밀번호(10자 이상), 회사·조직 이름(선택), 사용 목적(선택)을 적고 신청합니다.
- 운영자가 승인하면 로그인할 수 있습니다. 승인 전에 로그인하면 "가입 승인 대기 중"이라고 나옵니다.
- 승인되면 적은 회사·조직 이름으로 조직이 만들어지고, 신청한 사람이 그 조직의 소유자가 됩니다. 조직 이름은 나중에 콘솔의 조직 제목 옆 이름 변경에서 바꿀 수 있습니다 (소유자만).

비밀번호와 이름은 로그인 후 왼쪽 아래 내 계정 뱃지(이름) → 내 정보 / 비밀번호 변경에서 바꿉니다. 비밀번호를 바꾸면 다른 기기의 로그인은 끊기고, 내 정보의 다른 기기에서 모두 로그아웃으로 직접 끊을 수도 있습니다.

2단계. 콘솔에서 프로젝트 만들기 (2분)

- <https://fix.bebo.kr/console/> 에 로그인합니다.
- 새 프로젝트에 이름(예: NabiMan)과 사이트 주소(예: `https://man.bebo.kr`)를 넣고 만들기를 누릅니다.
 - 주소는 `https://도메인` 형식만 됩니다 (경로 없이). 여러 개면 한 줄에 하나씩 적습니다.
 - 여기 적은 사이트에서만 위젯 창이 열립니다.
- 만들어진 카드에서 3. Claude Code 반영 방식을 고릅니다.

방식	동작	권장 대상
계획 후 승인 (<code>plan</code> , 기본)	AI가 원인과 계획을 리포트 창에 올리고, 승인 답을 받은 뒤 고칩니다	처음 도입, 서버 관리처럼 위험도가 큰 서비스
브랜치·PR 작성 (<code>pr</code>)	'자동 처리 허용' 리포트는 새 브랜치에서 고치고 커밋합니다. 반영은 사람이 합니다	git으로 관리하는 서비스
바로 반영·배포 (<code>direct</code>)	'자동 처리 허용' 리포트는 테스트 후 바로 배포합니다	충분히 검증된 뒤

3단계. 사이트에 스크립트 한 줄 붙이기 (3분)

카드의 **1. 사이트에 붙일 코드**를 복사해 사이트 HTML의 `</body>` 바로 앞에 붙입니다.

5단계의 AI 설치 프롬프트나 설치 한 줄을 쓰면 이 단계는 건너뛰어도 됩니다. AI가 스크립트를 붙이는 계획을 제안하고, 승인하면 반영합니다.

```
<script src="https://fix.bebo.kr/w.js" data-site="fs_여기에_사이트_키" async></script>
```

- 화면 오른쪽 아래에 **개선 리포트** 버튼이 생기면 성공입니다.
- NabiMan(React)의 경우 `frontend/public/index.html` 의 `</body>` 앞에 넣고, 평소 방식으로 빌드·배포합니다.
- 화면 맥락을 더 알려 주고 싶으면(선택) 사이트 코드에서 다음처럼 호출합니다. 값은 리포트에 함께 저장됩니다.

```
window.FixHub?.setContext({ server: "web-01", menu: "services" });
```

- 주소창의 `token`, `password`, `key`, `session`, `otp` 같은 값은 자동으로 빠집니다.

4단계. 리포트를 쓸 사람 초대하기 (1분)

콘솔의 **사람 초대**에 이메일을 넣고 **초대 링크 만들기**를 누른 뒤, 나온 링크를 그 사람에게 보냅니다.

- 리포터: 리포트 작성, 대화
- 관리자: 프로젝트 설정, 에이전트 키, 리포터 초대·내보내기
- 소유자: 조직 전체 (구성원 역할 변경 포함)
- 조직 제목 옆 **구성원**에서 역할 변경, 내보내기, 사용 전 초대 취소를 할 수 있습니다.
- 링크는 7일 동안 한 번만 쓸 수 있습니다. 링크를 연 사람은 이름과 비밀번호만 정하면 됩니다.

5단계. Claude Code(또는 Codex) 연결하기 (1분)

카드의 **4. Claude Code 연결** → **설치 한 줄 만들기**를 누르면 **AI 설치 프롬프트**와 **터미널 한 줄**이 함께 나옵니다. 둘 다 10분 동안 한 번만 쓸 수 있는 연결 코드가 들어 있으니 **둘 중 하나만** 쓰십시오. 에이전트 키는 어디에도 표시되지 않고, 서버의 권한 600 파일에만 저장됩니다.

방법 A. AI에게 말하기 (권장, Claude Code 또는 Codex가 설치된 서버)

1. 사이트 소스 폴더에서 Claude Code(`claude`) 또는 Codex(`codex`)를 엽니다.
2. 복사한 **AI 설치 프롬프트**를 붙여 넣습니다. 프롬프트는 이런 모양입니다.

```
fixhub(https://fix.bebo.kr) 개선 리포트 위젯을 이 프로젝트에 설치해 줘. 대상 사이트: https://man.bebo.kr
1. 이 저장소의 최상위 폴더에서 아래 한 줄을 실행해. (연결 코드: 10분, 1회용)
curl -fsSL https://fix.bebo.kr/api/install.sh | sh -s -- FXC-XXXX-XXXX-XXXX
2. 공통 레이아웃의 </body> 앞에 위젯 스크립트를 넣는 계획을 보여 줘 (CSP가 있으면 함께).
3. .fix/config.json 의 build-test-deploy-protected 를 채우는 안을 보여 줘.
4. 2~3번은 내가 승인한 뒤에만 반영하고, 빌드-테스트 결과를 알려 줘.
5. 마지막으로 fixhub status 결과와 다음 할 일을 알려 줘.
```

3. AI가 설치를 마치고 **위젯 스크립트·CSP·빌드 명령 계획**을 보여 주면 확인하고 승인합니다. 3단계(스크립트 붙이기)를 AI가 대신 합니다.
4. 마무리
 - Claude Code: `/fix:watch` 를 입력하면 감시가 시작됩니다.

- Codex: \$fixhub 미처리 리포트를 확인하고 처리해 줘 라고 요청합니다. 설치 시 프로젝트의 .agents/skills/fixhub/ 에 스킬과 CLI가 준비됩니다. 스킬이 안 보이면 Codex를 다시 엽니다.
- Codex의 실행 환경에 따라 네트워크와 ~/.config/fixhub, ~/.local 쓰기에 승인이 필요할 수 있습니다. 실제 권한 안내를 확인하십시오. 설치 승인은 사이트 코드 수정이나 배포 승인과 별개입니다.

방법 B. 터미널 한 줄

1. Claude Code 또는 Codex를 쓰는 서버의 터미널에서 **사이트 소스 폴더로 이동한 뒤 복사한 터미널 한 줄을 붙여 넣습니다.**

```
cd ~/my-site
curl -fsSL https://fix.bebo.kr/api/install.sh | sh -s -- FXC-XXXX-XXXX-XXXX
```

- fixhub 명령, .fix/config.json, 에이전트 키와 사용할 AI의 확장 기능을 설치하고 연결을 확인합니다.
 - 기본값은 설치된 CLI를 감지합니다. 두 CLI가 있으면 둘 다 준비하며, 둘 다 없으면 Codex 스킬을 준비합니다.
 - Codex만 설치하려면 명령 끝에 --client codex, Claude Code만은 --client claude, 둘 다는 --client both를 붙입니다. Claude Code 플러그인 설치에는 claude 명령이 필요합니다.
 - 필요한 것: git, python3 (3.8 이상). 스크립트 내용은 <https://fix.bebo.kr/api/install.sh> 에서 먼저 읽어 볼 수 있습니다.
2. 같은 폴더에서 Claude Code는 /fix:watch, Codex는 \$fixhub 를 사용합니다. 위젯 스크립트와 프로젝트의 build/test/deploy 설정도 완료해야 합니다.

방법 C. Claude Code 직접 설치 (한 줄을 쓰지 않을 때)

1. 카드의 **4. Claude Code 연결 → 에이전트 키 발급**을 누릅니다. 명령 두 개가 한 번만 표시됩니다.
2. Claude Code를 쓰는 서버의 터미널에서, **사이트 소스 폴더로 이동한 뒤 두 명령을 차례로 붙여 넣습니다.**

```
cd ~/my-site # 예: 사이트 소스 폴더
claude plugin marketplace add https://fix.bebo.kr/plugin.git
claude plugin install fix@fixhub --scope project
mkdir -p ~/.config/fixhub && (umask 077; printf '%s' 'fa_...' > ~/.config/fixhub/fs_....key)
```

- 키는 채팅창에 붙이지 말고 반드시 이 명령으로 파일에 저장합니다 (본인만 읽을 수 있는 파일).
3. 같은 폴더에서 Claude Code를 열고 입력합니다.

```
/fix:watch
```

- 직접 설치했다면 처음 한 번은 Claude가 사이트 키를 물어 .fix/config.json 을 만들고, 빌드·테스트 명령을 채우도록 안내합니다. 이 파일에는 비밀 정보가 없으니 저장소에 커밋해도 됩니다.
- 오른쪽 아래 버튼의 점이 초록색이 되고, 위젯 창 맨 위에 ● **Claude Code 연결됨** (… · 서버 이름 · 시작 시각)이 보이면 끝입니다.
- 확실히 하려면 위젯 창 맨 위(또는 콘솔 프로젝트 카드)의 **연결 확인**을 누릅니다. 약 15초 안에 "응답 확인 — 왕복 N초"가 나오면 Claude Code와 서로 주고받는 것이 확인된 것입니다 (AI 사용량은 들지 않습니다).

Claude Code를 다른 폴더에서 열 때

.fix/config.json 은 저장소 최상위에 생깁니다. 그보다 위 폴더에서 Claude Code를 열어야 한다면 그 폴더의 .claude/settings.local.json 에 위치를 적어 두면 /fix:watch 가 자동으로 연결됩니다.

```
{ "env": { "FIXHUB_PROJECT": "/절대/경로/사이트-소스-폴더" } }
```

Codex 스킬 설치·업데이트와 사용

이미 fixhub에 연결한 프로젝트

연결 코드를 다시 발급할 필요가 없습니다. 아래 배포 ZIP에는 스킬과 CLI가 함께 들어 있습니다.

- [Codex 배포 ZIP 내려받기](#)
- [SHA-256 확인 파일](#)

ZIP을 풀고 `fixhub-codex` 폴더에서 실행합니다. 프로젝트 경로는 실제 사이트 소스 폴더로 바꿉니다.

```
python3 codex/install.py --project /absolute/path/to/project
```

프로젝트의 `.agents/skills/fixhub/`에 설치됩니다. 기존 `.fix/config.json`, 에이전트 키와 다른 스킬은 바꾸지 않습니다. 업데이트도 새 ZIP을 풀고 같은 명령을 실행합니다. 직접 수정한 설치 파일이 있으면 덮어쓰지 않고 중단하므로 변경분을 별도로 보관한 뒤 업데이트하십시오. 처음 연결하는 프로젝트는 앞의 한 줄 설치를 먼저 사용하십시오.

리포트 처리와 승인 후 이어가기

1. 사이트 소스 폴더에서 Codex를 열고 `$fixhub` 미처리 리포트를 확인하고 처리해 줘 라고 요청합니다.
2. 특정 리포트만 처리하려면 `$fixhub 3번` 리포트를 확인해 줘 처럼 번호를 지정합니다.
3. `plan` 방식에서는 Codex가 위젯 대화에 계획을 남깁니다. 위젯에서 승인 답변을 쓴 뒤 `$fixhub 3번` 승인 답변을 확인하고 이어서 처리해 줘 라고 다시 요청합니다.
4. 결과에는 수정 내용·테스트 결과·실제 배포 여부가 기록됩니다. PR 준비와 운영 배포는 구분해서 표시합니다.

프로젝트 최상위에서 다음 명령으로 스킬에 포함된 CLI를 직접 확인할 수도 있습니다.

```
python3 .agents/skills/fixhub/bin/fixhub status
python3 .agents/skills/fixhub/bin/fixhub reports
```

다른 폴더에서 작업해야 한다면 `FIXHUB_PROJECT`를 프로젝트의 절대 경로로 설정할 수 있습니다. 스킬 자동 검색을 위해 Codex는 사이트 프로젝트 안에서 여는 것을 권장합니다.

자동 감시와의 차이

이 Codex 스킬은 호출할 때 한 번 처리합니다. 새 리포트나 승인 답변이 와도 자동으로 다시 실행되지 않습니다. 따라서 `status`와 `reports`가 정상이어도 감시 표시가 회색일 수 있습니다. 화면의 **Claude Code 연결됨**과 **연결 확인**은 상주 감시 프로세스의 상태이며, Codex 스킬 설치 여부를 나타내지 않습니다.

`fixhub watch`는 변경 사항을 출력하고 연결 신호를 보내는 CLI입니다. 이것만 실행해도 Codex가 자동으로 코드를 고치는 것은 아닙니다. 자동 처리는 별도 감시 프로그램에서 `codex exec`를 호출하도록 확장할 수 있지만, 이번 배포에는 상주 실행기·자동 승인·자동 배포 예약을 포함하지 않습니다.

공식 문서: [Codex 스킬](#), [Codex 스크립트 실행](#).

쓰는 법

하고 싶은 일	방법
리포트 쓰기	버튼 → 새 리포트 → 제목·내용 → (선택) 요소 지정으로 문제 부분 클릭 → 등록
AI가 먼저 묻게 하기	리포트를 쓸 때 의견 피드백 체크
Claude 답 보기·답하기	목록 → 리포트 선택 → 아래 Claude Code 대화. 10초마다 새 답이 보입니다. 답변 필요 표시가 있으면 답을 기다리는 중입니다
리포트와 상관 없이 묻기	Claude 대화 탭
리포트 찾기	목록에서 상태·페이지 선택, 이 페이지만, 검색. 자동 은 AI 자동 처리 허용 리포트
문제 위치 다시 보기	리포트 상세의 작성 위치로 이동 → 그 페이지가 열리고 리포트와 지정한 요소가 강조됩니다
상태·설정 바꾸기	리포트 상세에서 상태, AI 자동 처리 허용, 의견 피드백 변경 (삭제는 조직 소유자만)
사이트 메뉴에서 대화 열기	사이트 코드에서 FixHub.open("chat") 호출
플러그인 업데이트	서버의 플러그인이 예전 버전이면 위젯 상태 줄(또는 콘솔 카드)에 플러그인 업데이트 (현재 → 최신) 이 보입니다 (소유자·관리자). 누르면 서버의 감시 프로세스가 약 15초 안에 업데이트하고 결과를 보여 줍니다. 적용하려면 Claude Code를 다시 시작하십시오
한 번만 처리	Claude Code에서 /fix:run , Codex에서 \$fixhub

문제가 생기면

증상	확인할 것
Codex에서 \$fixhub 가 안 보임	사이트 프로젝트 안에서 Codex를 열었는지, .agents/skills/fixhub/SKILL.md 가 있는지 확인하고 Codex를 다시 엽니다
Codex 설치가 기존 파일 변경으로 중단됨	직접 수정한 스킬을 별도 폴더에 보관한 뒤 설치합니다. 다른 스킬이나 프로젝트 설정을 삭제하지 마십시오
버튼이 안 보임	스크립트 한 줄의 data-site 값, 브라우저 콘솔 오류, 사이트 CSP가 https://fix.bebo.kr 스크립트를 막는지 (script-src 에 추가)
위젯 창이 비어 있음	콘솔 2. 허용할 사이트 주소에 지금 사이트 주소가 정확히 있는지 (https:// , 포트 포함)
사이트 CSP에서 창이 막힘	사이트 CSP의 frame-src (또는 child-src)에 https://fix.bebo.kr 추가
"권한이 없습니다"	이 사이트의 조직에 초대받은 계정인지 확인
점이 회색 / "연결 안 됨"	그 서버에서 Claude Code가 켜져 있고 /fix:watch 를 실행했는지. 키를 다시 발급했다면 5 단계의 키 저장을 다시
/fix:watch 뒤에도 "연결 안 됨", 또는 "setup needed" 알림	Claude Code를 .fix/config.json 이 있는 폴더(또는 그 하위 폴더)에서 열었는지. 다른 폴더라면 .claude/settings.local.json 의 env 에 FIXHUB_PROJECT 지정 (5단계 참고)

증상	확인할 것
설치 한 줄·프롬프트가 "연결 코드가 맞지 않거나 만료"	코드는 10분, 한 번만 유효합니다. 콘솔에서 설치 한 줄 만들기 를 다시 누르십시오
점이 노란색 / "응답 지연"	45초 넘게 신호가 없음. 서버·네트워크가 느리거나 Claude Code 세션이 막 끝났을 수 있음. 2분이 지나면 "연결 안 됨"으로 바뀝니다
"60초 안에 응답이 없습니다"	Claude Code 세션이 꺼졌거나 <code>/fix:watch</code> 가 실행되지 않은 상태. 세션을 다시 열고 <code>/fix:watch</code>
<code>marketplace add</code> 에서 "shallow capabilities" 오류	2026-09-26 이전 문제로 해결됨. 계속되면 운영자에게 알려 주십시오
키를 잃어버림	콘솔에서 다시 발급 (이전 키는 바로 막힙니다)
비밀번호를 잊어버림	운영자에게 요청하면 1회용 재설정 링크(24시간)를 받습니다
"정지된 계정입니다"	운영자에게 문의