

FlowNote 사용자 가이드 (v5.0)

FlowNote로 문서를 AI 기반으로 자동 분류하고 관리하는 방법을 안내합니다.

목차

1. 시작하기
2. 온보딩 가이드
3. 파일 분류
4. 키워드 검색
5. 대시보드 활용
6. Graph View 사용법
7. MCP & Obsidian 연동
8. 모바일 사용 가이드
9. 문제 해결
10. 팁 & 트릭

1. 시작하기

1.1 환경 요구사항

- **Python:** 3.11 이상
- **Node.js:** 18 이상 (Frontend용)
- **운영체제:** Windows, macOS, Linux
- **OpenAI API Key:** platform.openai.com
- **Redis:** 7.x (Celery 브로커용)

1.2 설치 및 실행

Step 1: 저장소 클론

```
git clone https://github.com/jjaayy2222/flownote-mvp.git  
cd flownote-mvp
```

Step 2: 가상환경 설정

```
# Unix/macOS:  
source venv/bin/activate # 또는 pyenv activate <your-env>  
# Windows:  
# venv\Scripts\activate  
  
pip install -r requirements.txt
```

Step 3: 환경 변수 설정

프로젝트 루트에 `.env` 파일 생성:

```
# OpenAI API Keys
OPENAI_API_KEY=sk-your-api-key-here

# GPT-4o (분류 전용)
GPT40_API_KEY=sk-your-api-key-here
GPT40_BASE_URL=https://api.openai.com/v1
GPT40_MODEL=gpt-4o

# GPT-4o-mini (경량 작업)
GPT40_MINI_API_KEY=sk-your-api-key-here
GPT40_MINI_BASE_URL=https://api.openai.com/v1
GPT40_MINI_MODEL=gpt-4o-mini

# Embeddings
EMBEDDING_API_KEY=sk-your-api-key-here
EMBEDDING_BASE_URL=https://api.openai.com/v1
EMBEDDING_MODEL=text-embedding-3-small

# Redis (Celery)
REDIS_URL=redis://localhost:6379/0

# Obsidian (Optional)
OBSIDIAN_VAULT_PATH=/path/to/your/vault
OBSIDIAN_SYNC_ENABLED=true
OBSIDIAN_SYNC_INTERVAL=300
```

Step 4: Redis 설치 및 실행

```
# macOS
brew install redis
brew services start redis

# Ubuntu/Debian
sudo apt-get install redis-server
sudo systemctl start redis

# Windows
# https://redis.io/docs/getting-started/installation/install-redis-on-windows/
```

Step 5: Frontend 의존성 설치

```
cd web_ui  
npm install  
cd ..
```

Step 6: 전체 서비스 실행

Terminal 1 - FastAPI Backend:

```
# 프로젝트 루트 디렉토리에서  
source venv/bin/activate  
python -m uvicorn backend.main:app --reload  
# → http://127.0.0.1:8000
```

Terminal 2 - Next.js Frontend:

```
cd web_ui  
npm run dev  
# → http://localhost:3000
```

Terminal 3 - Celery Worker & Beat:

```
celery -A backend.celery_app.celery worker --beat --loglevel=info
```

Terminal 4 - Flower (모니터링):

```
celery -A backend.celery_app.celery flower --port=5555  
# → http://localhost:5555
```

Terminal 5 - MCP 서버 (Optional):

```
# Claude Desktop 연동 시  
python -m backend.mcp.server
```

Step 7: 브라우저에서 확인

- Next.js 대시보드: <http://localhost:3000>
- FastAPI Docs: <http://127.0.0.1:8000/docs>
- Flower 모니터링: <http://localhost:5555>

2. 🚀 온보딩 가이드

2.1 온보딩이란?

온보딩은 FlowNote가 당신의 직업과 관심 영역을 학습하여 맞춤형 분류를 제공하기 위한 초기 설정 과정입니다.

2.2 온보딩 단계

Step 1: 대시보드 접속

1. <http://localhost:3000> 접속
2. 첫 방문 시 온보딩 프롬프트 표시

Step 2: 기본 정보 입력

1. 이름 입력 (예: **Jay**)
2. 직업 입력 (예: **개발자**, **디자이너**, **교사**)
3. 다음 → 버튼 클릭
4. GPT-4o가 당신의 직업에 맞는 **10개** 영역 추천

Step 3: 관심 영역 선택

1. 추천된 10개 영역 중 **정확히 5개** 선택
2. 선택 예시:
 - Python Development
 - Machine Learning
 - Web Development
 - Data Science
 - Project Management
3. 완료 버튼 클릭

Step 4: 완료 확인

- 온보딩 완료!
- 사용자 정보 확인:
 - 이름
 - 직업
 - User ID
 - 선택한 5개 영역

3. 💾 파일 분류

3.1 파일 업로드

단계별 안내

1. 대시보드 메인 페이지 이동
2. 파일 업로드 버튼 클릭

3. 파일 선택 (PDF, TXT, MD 지원)

4. 파일 정보 확인:

- 파일명
- 파일 크기
- 파일 타입

지원 파일 형식

- PDF (.pdf)
- TXT (.txt)
- Markdown (.md)

3.2 자동 분류 실행

1. **분류 시작** 버튼 클릭
2. AI 분석 진행 중... (사용자 맥락 반영)
3. 분류 결과 확인:

- **카테고리:** Projects/Areas/Resources/Archives
- **신뢰도:** 0-100%
- **키워드:** 추출된 주요 키워드
- **분류 근거:** AI의 판단 이유

3.3 분류 결과 이해하기

PARA 카테고리 설명

-  **Projects** (프로젝트)
→ 기한/마감일이 있는 구체적 목표
→ 예: "11월 30일까지 대시보드 구현"
→ 신호: "마감일", "프로젝트", "완료", "목표"

-  **Areas** (분야)
→ 지속적 책임 영역
→ 예: "팀 성과 관리는 계속 진행"
→ 신호: "지속", "관리", "모니터링", "유지"

-  **Resources** (자료)
→ 참고용 정보/학습 자료
→ 예: "Python 최적화 가이드"
→ 신호: "가이드", "참고", "문서", "학습"

-  **Archives** (보관)
→ 완료된 프로젝트 보관
→ 예: "2024년 프로젝트 결과"
→ 신호: "완료", "종료", "보관", "과거"

신뢰도 점수 해석

- **90-100%**: 매우 확실한 분류
 - **70-89%**: 높은 신뢰도
 - **50-69%**: 중간 신뢰도 (수동 확인 권장)
 - **50% 미만**: 낮은 신뢰도 (재분류 권장)
-

4. 🔎 키워드 검색

4.1 검색 실행

1. 대시보드 상단 검색바 사용
2. 검색어 입력 (예: [프로젝트 목표](#))
3. Enter 또는 검색 버튼 클릭

4.2 검색 결과 확인

- 파일명, 카테고리, 신뢰도 표시
- 유사도 점수 기반 정렬
- 검색 결과 미리보기

4.3 고급 검색 (FAISS)

- 벡터 기반 의미 검색
 - OpenAI Embeddings 활용
 - 실시간 유사도 점수 표시
-

5. 📈 대시보드 활용

5.1 메인 대시보드 (/)

Sync Monitor

- Obsidian 연결 상태 확인
- MCP 서버 상태 모니터링
- 최근 동기화 시간 표시
- 충돌 이력 확인

Analytics Overview

- PARA 분류 통계
- 파일 분포 차트
- 최근 활동 로그

5.2 Statistics 페이지 (/stats)

Activity Heatmap

- GitHub 스타일 연간 활동 히트맵
- 파일 생성/수정 빈도 시각화

- 일별 활동량 확인

Weekly Trend

- 최근 12주간 파일 처리량
- Line Chart로 추이 확인
- 주간 평균 계산

PARA Distribution

- 카테고리별 파일 비중
- Pie Chart 시각화
- 실시간 업데이트

6. Graph View 사용법

6.1 Graph View 접속

1. 좌측 사이드바에서 **Graph View** 클릭
2. 또는 <http://localhost:3000/graph> 직접 접속

6.2 그래프 조작

Zoom & Pan

- 마우스 휠: 줌 인/아웃
- 드래그: 그래프 이동
- Controls: 우측 하단 컨트롤 버튼 사용

노드 인터랙션

- 노드 클릭: 파일/카테고리 정보 Toast 알림
- 노드 타입:
 - 큰 원: PARA 카테고리 (Projects, Areas, Resources, Archives)
 - 작은 원: 개별 파일

MiniMap

- 우측 하단 미니맵으로 전체 구조 파악
- 현재 뷰포트 위치 확인

6.3 그래프 해석

- 엣지(연결선): 파일이 속한 카테고리 표시
- 노드 위치: Deterministic Layout (새로고침 시에도 유지)
- 색상: 카테고리별 구분

7. MCP & Obsidian 연동

7.1 Obsidian 동기화 설정

Step 1: Vault 경로 설정

.env 파일에 Vault 경로 추가:

```
OBSIDIAN_VAULT_PATH=/Users/your-name/Documents/ObsidianVault  
OBSIDIAN_SYNC_ENABLED=true  
OBSIDIAN_SYNC_INTERVAL=300
```

Step 2: 동기화 시작

1. Backend 서버 재시작
2. Sync Monitor에서 연결 상태 확인
3. Vault 내 파일 자동 감지 및 분류

7.2 Claude Desktop 연동 (MCP)

Step 1: 설정 파일 생성

```
# macOS  
cp claude_desktop_config.example.json ~/Library/Application\  
Support/Claude/claude_desktop_config.json  
  
# 설정 파일 편집  
# - 프로젝트 경로 수정  
# - Vault 경로 수정
```

Step 2: Claude Desktop 재시작

Step 3: MCP Tools 사용

Claude에게 다음과 같이 요청:

- "내 노트에서 '프로젝트' 관련 내용 찾아줘"
- "이 텍스트를 분류해줘: [텍스트]"
- "자동화 통계 보여줘"

7.3 충돌 해결

충돌 감지

- 3-way 충돌 감지 (local vs remote vs last_synced)
- Sync Monitor에서 충돌 이력 확인

자동 해결

- Rename 전략: 충돌 파일 백업 생성
 - 파일명: `filename_conflict_timestamp.md`
-

8. 모바일 사용 가이드

8.1 모바일 접속

1. 모바일 브라우저에서 `http://localhost:3000` 접속 (또는 배포된 URL)
2. 자동으로 모바일 레이아웃 전환

8.2 모바일 내비게이션

햄버거 메뉴

- 좌측 상단 햄버거 아이콘(☰) 클릭
- 슬라이드 드로어 메뉴 열림
- 메뉴 항목:
 - Dashboard
 - Graph View
 - Statistics
 - Preferences
 - GitHub

메뉴 닫기

- 메뉴 외부 영역 클릭
- 또는 X 버튼 클릭

8.3 모바일 최적화 기능

- 반응형 그리드: 1열 레이아웃 자동 전환
 - 터치 친화적: 버튼 크기 최적화
 - 스크롤 최적화: 가로 스크롤 없음 (375px 기준)
 - Drawer Scroll: 메뉴가 길 경우 스크롤 지원
-

9. 문제 해결

9.1 자주 발생하는 오류

OPENAI_API_KEY not found

원인:

- `.env` 파일 누락
- 환경 변수 설정 오류

해결:

```
# .env 파일 생성  
cp .env.example .env  
# API Key 입력 후 서버 재시작
```

✖ Redis connection failed

원인:

- Redis 서버 미실행

해결:

```
# macOS  
brew services start redis  
  
# Ubuntu/Debian  
sudo systemctl start redis  
  
# 연결 확인  
redis-cli ping  
# 응답: PONG
```

✖ Port 3000 already in use

원인:

- 다른 프로세스가 포트 사용 중

해결:

```
# 프로세스 확인 및 종료  
lsof -ti:3000 | xargs kill -9  
  
# 또는 다른 포트 사용  
npm run dev -- -p 3001
```

✖ MCP 서버 연결 실패

원인:

- Claude Desktop 설정 오류
- 경로 불일치

해결:

1. `claude_desktop_config.json` 경로 확인
2. 프로젝트 절대 경로 확인

3. Claude Desktop 재시작

9.2 성능 최적화 팁

파일 전처리

- 불필요한 페이지 제거
- 텍스트 품질 확인
- 이미지는 OCR 처리

검색 최적화

- 명확한 키워드 사용
- 너무 긴 질문 피하기
- 핵심 단어 포함

리소스 관리

- 불필요한 파일 삭제
- 주기적으로 캐시 정리
- 메모리 사용량 모니터링

10. 팁 & 트릭

10.1 효율적인 검색법

 나쁜 예:

"이 프로젝트에서 우리가 달성하고자 하는 목표가 무엇인지 자세히 알려주세요"

 좋은 예:

"프로젝트 목표"

10.2 파일명 규칙

- 명확한 이름 사용
- 날짜 포함 (예: **2025-11-project-report.pdf**)
- 버전 표시 (예: **budget_v2.pdf**)

10.3 분류 정확도 향상

1. 온보딩 정보를 정확하게 입력

- 직업을 구체적으로 작성
- 관심 영역을 신중하게 선택

2. 파일명을 명확하게 작성

- 내용을 유추할 수 있는 파일명 사용
- 카테고리를 암시하는 키워드 포함

3. 신뢰도가 낮으면 재분류

- 50% 미만 신뢰도는 재분류 권장
- 파일 내용 보완 후 재업로드

10.4 생산성 향상 워크플로우

1. 온보딩 완료 (최초 1회)
2. Obsidian Vault 연동 (자동 동기화)
3. 파일 생성 시 자동 분류
4. Graph View로 구조 파악
5. Stats로 활동 추이 확인
6. 필요 시 키워드 검색
7. Claude Desktop으로 AI 어시스턴트 활용

10.5 자동화 활용

Celery 스케줄러

- 매일 00:00: 자동 재분류 (신뢰도 낮은 파일)
- 매주 일요일: 스마트 아카이빙 (90일 이상 미수정 파일)
- 매주 월요일: 주간 리포트 생성

Flower 모니터링

- <http://localhost:5555> 접속
- 작업 상태 실시간 확인
- 워커 성능 모니터링

11. ☎ 지원 및 문의

문제가 해결되지 않나요?

- GitHub Issues: [이슈 등록](#)
- 작성자: [@jjaayy2222](#)
- 문서: [README.md](#) 참조

12. 🔄 업데이트 내역

v5.0.0 (2026-01-06) - 현재 버전

- MCP 서버 & Obsidian 동기화
- Next.js 기반 모던 대시보드
- PARA Graph View (React Flow)

- Advanced Stats (Recharts)
- Mobile Responsive UI
- Accessibility 개선 (ARIA, 스크린 리더)

v4.0 (2025-12-16)

- Celery 기반 자동화 시스템
- Redis 통합
- Flower 모니터링

v3.5 (2025-11-11)

- 스마트 온보딩 (GPT-4o 영역 추천)
 - 맥락 기반 분류
 - 실시간 대시보드 (Streamlit)
-

FlowNote v5.0을 사용해 주셔서 감사합니다! 🎉

더 나은 경험을 위해 지속적으로 개선하고 있습니다.

피드백은 언제나 환영합니다! ❤️

Made with ❤️ by [Jay](#)