# 운영 기록 증빙 — 파일 기반 이기종 AI 협업 체계

공유 폴더에 남은 실제 기록에서 추출. 사람 이름과 컴퓨터 경로만 가리고 나머지는 원문 그대로 수록.

## 1. 야간 작업 시간표 (2026년 8월 28일 00:00~10:00)

공유 폴더에 작업 지시서와 결과 보고서가 생긴 시각(10분 단위). 지시서 발행 후 30분 안팎에 결과 보고서가 돌아오는 주기가 반복됨.




| 구간 | 작업 지시서 | 결과 보고서 |
|---|---|---|
| 00:00~10:00 합계 | 64 | 58 |

집계 기준: 이 표는 8월 28일 00:00~10:00 사이에 기록된 문서만 집계했습니다. 3절 날짜별 표의 8월 28일 건수는 10시 이후 작업을 포함한 그날 폴더 전체 건수이므로 이 표보다 많습니다.

## 2. 실제 문서 한 세트 — 지시 → 결과 → 보완 지시 → 보완 결과

> 개발 AI 2가 권한 판정 로직을 구현(시험 20건 통과)한 뒤, 검증 AI가 "목록·사전형 입력에서 예외 발생"을 찾아 보완 지시서를 발행. 개발 AI 2가 수정 후 시험 23건 통과로 보고. 네 문서의 기록 시각: 00:38 → 01:07 → 01:17 → 01:19.

**작업 지시서 1차 · 파일 기록 시각 2026-08-28 00:38:14**

```
# 01 개발 AI 2 — 동작권한판정 순수로직

상태: READY — 2026-08-28 00:37 PM 위임으로 재개
난이도: 중

## 목적

메뉴 표시와 무관하게 서버에서 read/write/approve를 판정한다. View와 DB에는 연결하지 않는다.

## 먼저 읽기

1. `[프로젝트]\분석설계\[설계]\TMS_표준제품_전체설계.md`
2. `[프로젝트]\분석설계\[설계]\TMS_P1_모듈권한_설계.md`
3. `[프로젝트]\분석설계\[설계]\TMS_모듈_카탈로그.md`
4. `[프로젝트]\분석설계\[설계]\TMS_전체설계_검증_결과.md`

## 소유 파일

- `[프로젝트]\apps\system\access_policy.py`
- `[프로젝트]\apps\system\test_access_policy.py`

## 구현
```

- 입력: 사용자/역할 매핑·활성, 모듈활성, can_read/write/approve, action
- 모르는 값과 조회실패는 차단
- HTTP method만으로 action을 추측하지 않음
- approve는 can_read AND can_approve, can_write는 강제하지 않음
- read/write/approve 외 action 차단
- 반환: allowed, reason_code
- 최소 코드: USER_MAPPING_MISSING, USER_INACTIVE, ROLE_MAPPING_MISSING, ROLE_INACTIVE, MODULE_DISABLED, PERMISSION_MISSING, UNKNOWN_ACTION
- superuser 숨은 우회는 넣지 않고 임의결정에 기록

## 필수 시험

- read만 허용, write/approve 차단
- write 허용
- read 없는 approve 차단
- 비활성 사용자·역할·모듈 차단
- None·unknown 차단
- 승인권한 매트릭스 표본

실행: `python manage.py test apps.system.test_access_policy`

## 금지

- settings.py, views.py, 메뉴, DB, 기존 미커밋 파일 수정
- 커밋, push

## 결과

`[공유 폴더]\20260828\개발AI2\02_results\01_개발 AI 2_동작권한판정_순수로직_결과.md`

### 결과 보고서 1차 · 파일 기록 시각 2026-08-28 01:07:34

# 01 개발 AI 2 — 동작권한판정 순수로직 결과

- 담당: 개발 AI 2
- 완료시각: 2026-08-28 01:06:57
- 판정: 완료

## 산출물 (소유 파일만 생성)

| 파일 | 내용 |
|---|---|
| `[프로젝트]\apps\system\access_policy.py` | 순수 판정 로직 (신규) |
| `[프로젝트]\apps\system\test_access_policy.py` | 자기시험 (신규) |

다른 파일·settings.py·views.py·메뉴·DB·기존 미커밋 파일은 손대지 않았다. 커밋/push 없음.

## 시험 실행 명령과 결과

```
.venv\Scripts\python.exe manage.py test apps.system.test_access_policy
```

```
Ran 20 tests in 0.001s
OK
Found 20 test(s).
Skipping setup of unused database(s): default.
System check identified no issues (0 silenced).
```

- 20건 전부 통과. DB 미사용(SimpleTestCase)으로 순수로직임을 실행으로 확인.
- 최초 `python manage.py test` 는 시스템 파이썬이라 무반응 → 프로젝트 venv(`.venv\Scripts\python.exe`)로 실행해야 함(운영 참고).

## 설계 이행 확인

- 입력: 사용자/역할 매핑·활성, 모듈활성, can_read/write/approve, action → `AccessContext` dataclass 로 명시.
- action 은 read/write/approve 만 유효, 그 밖은 `UNKNOWN_ACTION` 차단. HTTP method 로 추측하지 않음(action 을 호출측이 명시 전달).
- 모르는 값·조회실패(None)·False 는 전부 차단 (`is not True` 검사).
- approve = `can_read AND can_approve`. can_write 강제하지 않음.
- 요청 동작의 권한값만 판정 (read 요청에서 can_write=None 은 무영향).
- 반환: `AccessDecision(allowed, reason_code)`.
- 사유코드 8종: ALLOWED + USER_MAPPING_MISSING, USER_INACTIVE, ROLE_MAPPING_MISSING, ROLE_INACTIVE, MODULE_DISABLED, PERMISSION_MISSING, UNKNOWN_ACTION.

## 판정 순서 (구현 결정)

```
1. action 유효성      -> UNKNOWN_ACTION
2. 사용자 매핑 존재    -> USER_MAPPING_MISSING
3. 사용자 활성        -> USER_INACTIVE
4. 역할 매핑 존재      -> ROLE_MAPPING_MISSING
5. 역할 활성          -> ROLE_INACTIVE
6. 모듈 활성          -> MODULE_DISABLED
7. 동작별 권한값      -> PERMISSION_MISSING
```

주체 → 역할 → 모듈 → 권한 순. 앞 단계 차단이 우선(`PrecedenceTest` 로 검증).

## 시험 목록 (20건)

- ReadWriteApprove: read 전용 사용자의 read 허용·write/approve 차단, write 허용, approve=R+A, read 없는 approve 차단, approve 플래그 없는 approve 차단
- IrrelevantPermission: read 가 can_write=None 무시, write 가 can_read/approve 무시
- InactiveSubject: 사용자매핑누락/비활성, 역할매핑누락/비활성, 모듈비활성 각 사유코드
- UnknownAndNone: unknown action 6종, 권한값 None 차단(read/write/approve), 주체 플래그 None 차단
- Precedence: action>주체, 사용자>역할, 역할>모듈, 모듈>권한 우선순위
- ApproveMatrix: can_read × can_approve 6조합 표본

## 임의결정 (이유·영향·되돌리는 법)

1. **superuser 우회 미포함**
   - 이유: 검증 결과 §4· 작업 지시서 지시대로 "순수정책에 숨은 우회 금지". Django superuser 라도 이 로직은 동일하게 판정한다.
   - 영향: 관리자도 권한값이 없으면 차단됨. 별도 명시정책(예: 명시적 admin 역할 권한)이 확정되기 전까지 이 로직만으로는 우회 불가.
   - 되돌리는 법: 우회가 필요하면 이 모듈 밖의 상위 검증에서 명시적으로 처리하고 별도 결정으로 기록한다. 이 파일에 숨은 분기를 넣지 않는다.

2. **판정 순서에서 action 유효성을 최우선으로 둠**
   - 이유: 무엇을 판정할지(read/write/approve) 정해지지 않으면 이후 권한 비교가 의미 없음.
   - 영향: 잘못된 action 은 주체 상태와 무관하게 UNKNOWN_ACTION 을 받는다(정보노출 최소, 입력검증 우선).
   - 되돌리는 법: 주체 차단을 먼저 노출하려면 순서를 조정. 단 사유코드 문자열은 유지.

3. **입력을 dataclass(AccessContext)로 계약화**
   - 이유: View/DB 비의존 + 필드 누락을 호출 시점에 드러내기 위함. 순수 dict 대신 명시적 계약.
   - 영향: 호출측이 8개 필드를 채워 넘겨야 함(권한 3종은 기본 None).
   - 되돌리는 법: dict 수용이 필요하면 얇은 어댑터 함수를 추가(로직 본체는 불변).

## 다음 단계(이 작업 지시서 범위 밖, 참고)

검증 §6 착수 허용범위 기준 다음 파동 후보: ③은 완료. View/API 공통 검증(§11-3)는 이 순수로직을 물어 쓰는 구조 – 별도 작업 지시서 대기.

### 보완 지시서(검증 AI 발행) · 파일 기록 시각 2026-08-28 01:17:31

# 02 개발 AI 2 – 권한입력 안정차단 보완

상태: READY
난이도: 중

## 원인

1차 구현은 action이 list/dict일 때 집합 조회에서 예외가 난다. 권한 판정은 잘못된 입력도 서버 오류가 아니라 차단 결과로 끝나야 한다.

## 소유 파일

- `[프로젝트]\apps\system\access_policy.py`
- `[프로젝트]\apps\system\test_access_policy.py`

## 완료조건

- action이 문자열이 아니거나 빈 값이면 예외 없이 `UNKNOWN_ACTION`으로 차단한다.
- 사용자·역할·모듈 활성 입력은 `True/False/None`을 명시적으로 다룬다.
- 기존 권한 행렬을 유지한다.
- list/dict/숫자/None/공백 action 공격시험을 추가해 모두 통과한다.
- View·DB·settings.py는 수정하지 않는다.

## 결과

`[공유 폴더]\20260828\개발AI2\02_results\02_개발 AI 2_권한입력_안정차단_보완_결과.md`

### 보완 결과 보고서 · 파일 기록 시각 2026-08-28 01:19:11

# 02 개발 AI 2 — 권한입력 안정차단 보완 결과

- 담당: 개발 AI 2
- 완료시각: 2026-08-28 01:18:45
- 판정: 완료

## 결함 재현 (수정 전)

`action` 이 unhashable(list/dict)일 때 `action not in VALID_ACTIONS`(frozenset)에서 예외 발생.

```
[] -> EXCEPTION: TypeError unhashable type: 'list'
{} -> EXCEPTION: TypeError unhashable type: 'dict'
5  -> UNKNOWN_ACTION   (숫자는 hashable 이라 정상 차단)
None -> UNKNOWN_ACTION
'  ' -> UNKNOWN_ACTION
```

권한 판정은 잘못된 입력도 서버 오류가 아니라 차단 결과로 끝나야 하는데, list/dict 는 500 을 유발했다.

## 수정 (소유 파일만)

`access_policy.py` — action 검증 1줄:

```python
# 변경 전
if action not in VALID_ACTIONS:

# 변경 후 (isinstance 를 먼저 보아 not in 의 TypeError 를 막음)
if not isinstance(action, str) or action not in VALID_ACTIONS:
```

`isinstance(action, str)` 가 False 면 short-circuit 으로 `not in` 을 실행하지 않으므로 unhashable 입력에서도 예외 없이 `UNKNOWN_ACTION` 으로 차단된다. 빈 문자열·공백은 기존대로 정확매칭 실패로 UNKNOWN_ACTION.

## 수정 후 검증

```
[] -> UNKNOWN_ACTION
{} -> UNKNOWN_ACTION
5  -> UNKNOWN_ACTION
None -> UNKNOWN_ACTION
'  ' -> UNKNOWN_ACTION
('read',) -> UNKNOWN_ACTION
```

## 시험 실행 명령과 결과

```
.venv\Scripts\python.exe manage.py test apps.system.test_access_policy
```

```
Ran 23 tests in 0.001s
OK
Found 23 test(s).
System check identified no issues (0 silenced).
```

- 기존 20건 유지 + 보완 3건 추가 = 23건 전부 통과.

## 완료조건 대응

| 완료조건 | 대응 |
|---|---|
| action 이 문자열 아니거나 빈 값이면 예외 없이 UNKNOWN_ACTION | `isinstance` 선검사로 해결. `test_non_string_action_never_raises`, `test_blank_action_is_blocked` |
| 사용자·역할·모듈 활성 입력 True/False/None 명시 처리 | `is not True` 검사 유지 + `test_activation_flags_true_false_none_explicit` (5개 플래그 × True 통과 / False·None 차단) 추가 |
| 기존 권한 행렬 유지 | 권한 판정 로직 무변경. ApproveMatrix 등 기존 시험 그대로 통과 |
| list/dict/숫자/None/공백 action 공격시험 추가·통과 | `test_non_string_action_never_raises`(list/dict/[read]/{dict}/int/float/bool/tuple), `test_blank_action_is_blocked`(빈·공백 5종) |
| View·DB·settings.py 미수정 | 소유 파일 2개만 수정 |

## 추가한 시험 (3건)

- `test_non_string_action_never_raises`: `[]`, `{}`, `[read]`, `{"action":read}`, `5`, `0`, `3.14`, `True`, `False`, `("read",)` → 전부 UNKNOWN_ACTION, 예외 없음
- `test_blank_action_is_blocked`: `""`, `" "`, `"\t"`, `"\n"`, `"  "` → UNKNOWN_ACTION
- `test_activation_flags_true_false_none_explicit`: 5개 활성 플래그 각 True 통과 / False·None 차단

## 임의결정

- **`True`/`False` action 을 UNKNOWN_ACTION 으로 처리**: bool 은 str 이 아니므로 isinstance 검사에서 걸러진다. Python 에서 `True`/`False` 는 dict 키로는 유효하나 action 문자열이 아니므로 차단이 맞다. 되돌릴 필요 없음.
- 공백 action(`" "`)은 strip 후 매칭하지 않고 정확매칭 실패로 차단. 앞뒤 공백을 허용하면 오탈자 통과 위험이 있어 관대화하지 않음.

## 금지사항 준수

settings.py·views.py·메뉴·DB·기존 미커밋 파일 미접촉. 커밋/push 없음.

## 3. 날짜별 작업 지시서·결과 보고서 건수

| 날짜 | 작업 지시서 | 결과 보고서 |
|---|---|---|
| 2026-08-05 | 0 | 12 |
| 2026-08-20 | 3 | 2 |

| | | |
|---|---|---|
| 2026-08-24 | 2 | 1 |
| 2026-08-25 | 6 | 5 |
| 2026-08-26 | 100 | 100 |
| 2026-08-27 | 11 | 0 |
| 2026-08-28 | 172 | 152 |
| 2026-08-29 | 0 | 0 |
| 2026-08-31 | 1 | 0 |
| 2026-09-01 | 34 | 34 |
| 2026-09-02 | 12 | 12 |
| 2026-09-03 | 7 | 8 |
| 2026-09-04 | 2 | 4 |
| 2026-09-07 | 0 | 4 |
| 2026-09-08 | 4 | 5 |
| 2026-09-09 | 0 | 4 |
| 2026-09-21 | 0 | 2 |
| 2026-10-06 | 15 | 12 |
| 2026-10-09 | 0 | 0 |
| 2026-10-13 | 1 | 0 |
| 2026-10-15 | 0 | 0 |
| **합계(21일)** | **370** | **357** |

집계 기준: 날짜별 폴더 아래 작업 지시서 폴더·결과 보고서 폴더의 문서 파일 수.

## 4. 원본 보관

원본 기록(문서 840건)과 작성 시각 정보를 보관하고 있으며, 주최 측이 요청할 경우 원본 열람과 AI 활용 방식 설명을 제공할 수 있습니다.