항상 같은 회차를 추천하던 비서를 고치고, Hermes에서 실제 답을 받았습니다

지피터스 24기 · 2026-10-13 공개 실습 자료

생활형AI비서 보강 상태 — 2026-10-13

이전: 정해진 E02 추천 문장을 출력하는 정적 예시와 Hermes 도움말 확인까지만 있었다. 이번:

1. missing_inputs / state / priority / evidence를 읽어 추천을 계산하는 brief.py로 수정.
2. 현재 입력 E02 추천, E02 완료 후 추천 없음, 합성 입력에서 E01 조건 충족 후 E01 추천, 새로운 E04 우선순위 선택, 중복 ID 거절을 확인.
3. 실제 Hermes CLI에 현재 결과 JSON을 주어 한국어 브리핑을 받음. exit0, session 20261013_173836_ae1b74. 응답: 오늘 E02의 브라우저 결과를 원고에 반영하고, E01은 출발지·날짜, E03은 OS·시험 참가자 정보가 부족하다고 설명했다.

범위: 로컬 상태 계산+Hermes 모델 응답. 메신저 연결과 일정 예약 발송은 아직 하지 않았다. 합성 입력 검사5개는 실사용5회가 아니다. Hermes 호출의 toolsets=none 옵션은 알 수 없는 toolset 경고가 났다. 이 옵션을 공식 사용법이나 권한 차단 근거로 제시하지 않는다. 기존 계정·gateway·전역 설정은 수정하지 않았다.

실행: python3 brief.py episodes-v2.json 상태 테스트: python3 test_brief.py 원문 응답과 실행조건: ../reinforcement-20261013/life/hermes-answer.md

Hermes 실제 CLI 응답

오늘은 E02 「AI 더치페이 도구를 믿어도 될까?」에서 브라우저 계산 결과를 결과 원고에 반영하면 됩니다.
E01은 출발지와 여행 날짜가 입력되지 않아 막혀 있습니다.
E03은 운영체제와 시험 참가자 정보가 없어 진행할 수 없습니다.

실행: 2026-10-13 17:38 KST / exit 0 / session 20261013_173836_ae1b74. 범위: 공개 실습 JSON을 입력한 CLI 모델 응답. Telegram·Slack 송수신은 하지 않음. 모델명은 이 출력에서 확인하지 않아 기재하지 않음. 옵션 참고: toolsets=none은 알 수 없는 toolset 경고가 났다. 이 값을 공식 지원 옵션으로 안내하지 않으며, 이번 결과로 도구 권한 차단 기능을 입증하지 않는다.

상태 변경 검사 출력

공개 실습 소스 / 긴 줄은 지면에 맞춰 자동 줄바꿈됩니다. 코드 복사 시 줄바꿈을 확인하세요.


```
{
  "scope": "입력 상태 변경 로컬 검사. 실제 사용자 5회 이용이 아님",
  "checks": [
    {
      "case": "현재 공개 실습 상태",
      "expected": "E02",
      "actual": "E02",
      "pass": true
    },
    {
      "case": "E02 완료·다른 회차는 입력 부족",
      "expected": null,
      "actual": null,
      "pass": true
    },
    {
      "case": "합성 조건: 여행 필수 입력이 채워진 경우",
      "expected": "E01",
      "actual": "E01",
      "pass": true
    },
    {
      "case": "새로운 ID도 우선순위로 선택",
      "expected": "E04",
      "actual": "E04",
      "pass": true
    },
    {
      "case": "중복 ID 거절",
      "pass": true
    }
  ]
}
```

life/brief.py

공개 실습 소스 / 긴 줄은 지면에 맞춰 자동 줄바꿈됩니다. 코드 복사 시 줄바꿈을 확인하세요.

```
"""Choose an actionable public-study episode from supplied state; no LLM or send."""
import argparse,json
from pathlib import Path
from datetime import datetime
from zoneinfo import ZoneInfo

STATES={'idea','draft','review','approved','published','done'}
def choose(rows):
 if not isinstance(rows,list):raise ValueError('회차 목록이 필요합니다.')
 seen=set();ready=[];blocked=[];completed=[]
 for r in rows:
  if not isinstance(r,dict):raise ValueError('회차는 객체여야 합니다.')
  ident=r.get('id')
  if not isinstance(ident,str) or not ident.strip() or ident in seen:raise ValueError('ID 누락 또는 중복')
  seen.add(ident)
  if r.get('state') not in STATES:raise ValueError('알 수 없는 상태')
  missing=r.get('missing_inputs');evidence=r.get('evidence')
  if not isinstance(missing,list) or not isinstance(evidence,list) or any(not isinstance(v,str) or not v.strip() for v in missing+evidence):raise ValueError('누락 입력과 근거 목록을 확인해주세요.')
  if type(r.get('priority')) is not int:raise ValueError('우선순위는 정수여야 합니다.')
  if not isinstance(r.get('title'),str) or not isinstance(r.get('next_action'),str) or not r['next_action'].strip():raise ValueError('제목과 다음 행동이 필요합니다.')
  if r['state'] in {'published','done'}:completed.append(ident)
  elif missing:blocked.append({'id':ident,'needs':missing})
  else:ready.append(r)
 ready.sort(key=lambda r:(r['priority'],-len(r['evidence']),r['id']))
 recommended=None
 if ready:
  r=ready[0];recommended={'id':r['id'],'title':r['title'],'next_action':r['next_action'],'reason':f"필수 입력 누락 0개, 지정 우선순위 {r['priority']}, 기록된 근거 {len(r['evidence'])}개"}
 return {'recommended':recommended,'ready_ids':[r['id'] for r in ready],'blocked':blocked,'completed':completed,'rule':'완료 항목 제외 → 누락 입력 없는 항목 → 지정 우선순위 오름차순 → 근거 개수 내림차순 → ID','limits':'입력 상태는 제공된 기록을 따르며 근거의 사실성·실제 사용 효과·메신저 전달을 검증하지 않음'}

if __name__=='__main__':
 p=argparse.ArgumentParser();p.add_argument('input',nargs='?',default=str(Path(__file__).with_name('episodes-v2.json')));args=p.parse_args()

result=choose(json.loads(Path(args.input).read_text()));result['as_of']=datetime.now(ZoneInfo('Asia/Seoul')).isoformat(timespec='seconds');print(json.dumps(result,ensure_ascii=False,indent=2))
```

life/episodes-v2.json

공개 실습 소스 / 긴 줄은 지면에 맞춰 자동 줄바꿈됩니다. 코드 복사 시 줄바꿈을 확인하세요.


```
[
  {
    "id": "E01",
    "title": "AI 여행 일정은 실제로 움직일 수 있을까?",
    "question": "이동·예산·운영시간을 검증할 수 있는가",
    "state": "draft",
    "next_action": "출발지와 여행 날짜 확인",
    "missing_inputs": [
      "출발지",
      "여행 날짜"
    ],
    "evidence": [],
    "priority": 1
  },
  {
    "id": "E02",
    "title": "AI 더치페이 도구를 믿어도 될까?",
    "question": "분할 금액 합계가 원래 총액과 같은가",
    "state": "review",
    "next_action": "브라우저 계산 결과를 결과 원고에 반영",
    "missing_inputs": [],
    "evidence": [
      "../evidence/split-tests.json",
      "../evidence/browser-tests.json"
    ],
    "priority": 1
  },
  {
    "id": "E03",
    "title": "AI 실습 안내만으로 페이지를 만들 수 있을까?",
    "question": "초보자가 안내만 보고 이름 변경을 완료하는가",
    "state": "draft",
    "next_action": "사용 운영체제와 시험 참가자 확인",
    "missing_inputs": [
      "운영체제",
      "시험 참가자"
    ],
    "evidence": [],
    "priority": 1
  }
]
```

life/test_brief.py

공개 실습 소스 / 긴 줄은 지면에 맞춰 자동 줄바꿈됩니다. 코드 복사 시 줄바꿈을 확인하세요.

```
import json,copy
from pathlib import Path
from brief import choose
rows=json.loads(Path(__file__).with_name('episodes-v2.json').read_text());checks=[]
def test(name,data,expected):
 out=choose(data);actual=out['recommended']['id'] if out['recommended'] else None;assert actual==expected,
(name,actual,expected);checks.append({'case':name,'expected':expected,'actual':actual,'pass':True})
test('현재 공개 실습 상태',rows,'E02')
x=copy.deepcopy(rows);x[1]['state']='done';test('E02 완료·다른 회차는 입력 부족',x,None)
x[0]['missing_inputs']=[];test('합성 조건: 여행 필수 입력이 채워진 경우',x,'E01')
x=copy.deepcopy(rows);new=copy.deepcopy(x[1]);new.update(id='E04',title='합성: 새 회
차',priority=0);x.append(new);test('새로운 ID도 우선순위로 선택',x,'E04')
try:choose(rows+[rows[0]])
except ValueError:checks.append({'case':'중복 ID 거절','pass':True})
else:raise AssertionError('duplicate')
print(json.dumps({'scope':'입력 상태 변경 로컬 검사. 실제 사용자 5회 이용이 아
님','checks':checks},ensure_ascii=False,indent=2))
```

before/brief.py

공개 실습 소스 / 긴 줄은 지면에 맞춰 자동 줄바꿈됩니다. 코드 복사 시 줄바꿈을 확인하세요.

```
import json
from pathlib import Path
p=Path(__file__).parent
rows=json.loads((p/'episodes-current.json').read_text())
print('# 연재 다음 행동 브리핑 · 2026-10-12')
print('공개 실습 자료 기준. 일정 추정 및 메신저 발송 없음.\n')
for r in rows: print(f"- {r['id']} {r['title']} → {r['next_action']}")
print('\n추천 착수: E02. 로컬 계산 실험을 수행했으므로 그 결과를 원고에 반영할 수 있습니다.')
print('E01은 여행 날짜·출발지가, E03은 운영체제·실제 시험 참가자가 필요합니다.')
```