

AEGIS AI Logic

Gemini API 통합 가이드 v2.0.0

Generated: 2025-12-04 06:05:40

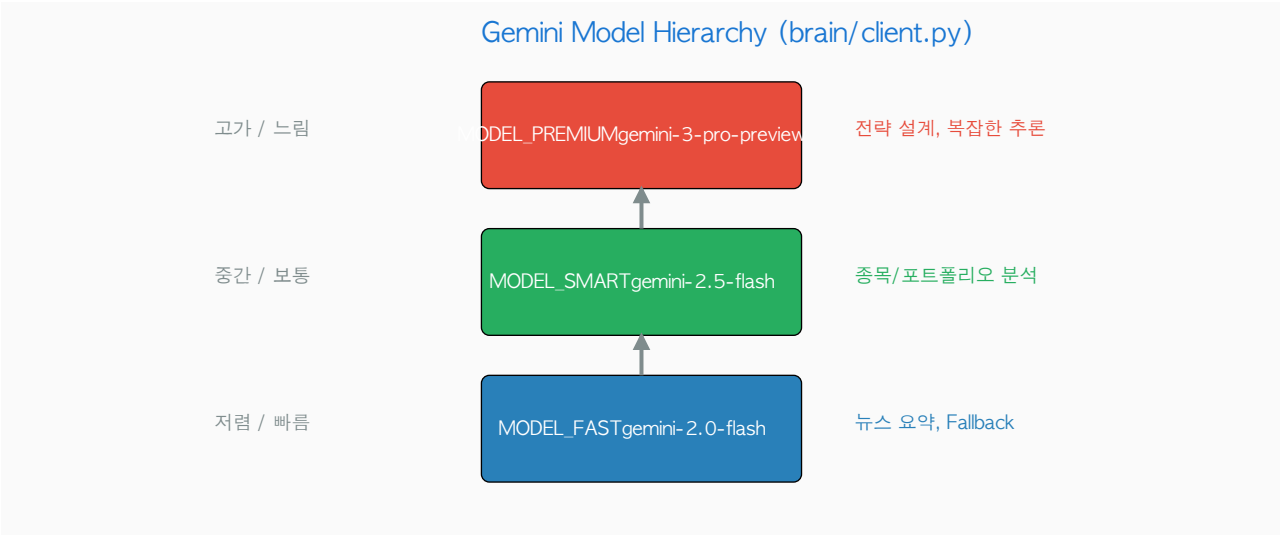
핵심 파일: brain/client.py | 테이블: ai_analysis

1. 개요

AEGIS 시스템에서 Gemini API를 활용한 AI 분석 시스템의 구조와 사용법을 정리합니다. 모든 AI 분석 결과는 ai_analysis 테이블에 저장되며 캐시로 재사용됩니다.

프로세스 파일	역할	대상 테이블
brain/client.py	Gemini API 클라이언트 (핵심)	-
brain/ai_cache.py	AI 분석 캐시 시스템	ai_analysis
brain/prompts.py	AI 프롬프트 템플릿	-
brain/integrated_analysis.py	통합 분석 모듈	ai_analysis
brain/strategy_architect.py	전략 설계 AI	ai_analysis
database/models.py	AIAnalysis 테이블 정의	ai_analysis
reporting/*.py	PDF 리포트에서 AI 호출	ai_analysis
fetchers/naver_finance/news.py	뉴스 테마 분석	ai_analysis, news

2. Gemini 모델 등급 (brain/client.py)



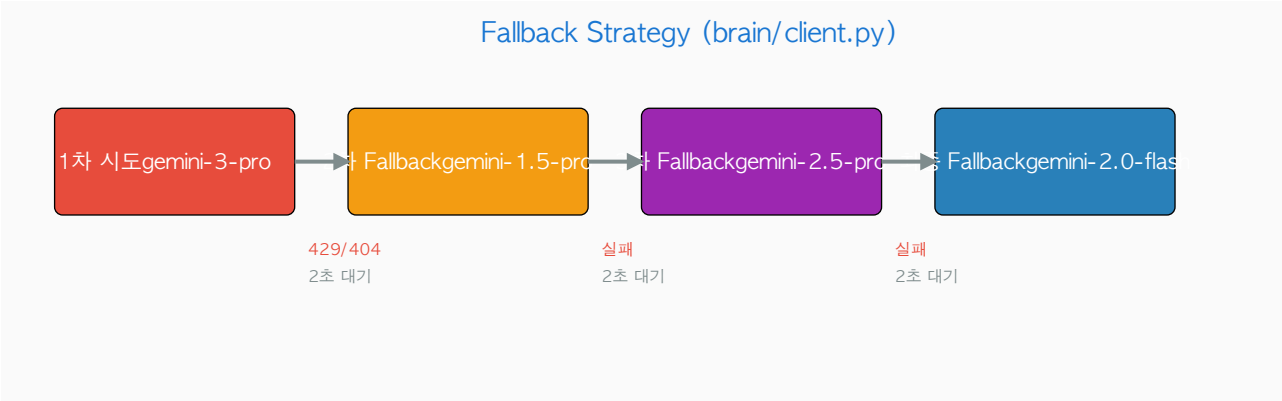
상수명	모델 ID	용도	비용	속도
MODEL_FAST	gemini-2.0-flash	뉴스 요약, 데이터 정제, Fallback	저렴	빠름
MODEL_SMART	gemini-2.5-flash	종목 분석, 포트폴리오 분석	중간	보통
MODEL_PREMIUM	gemini-3-pro-preview	전략 설계, 복잡한 추론, 심층 분석	고가	느림

3. 작업별 모델 매핑 (MODEL_BY_TASK)

작업 유형	모델	설명	캐시 시간
stock_analysis	MODEL_SMART	개별 종목 분석	24시간
portfolio_analysis	MODEL_SMART	포트폴리오 분석	1시간
strategy_design	MODEL_PREMIUM	전략 설계	6시간

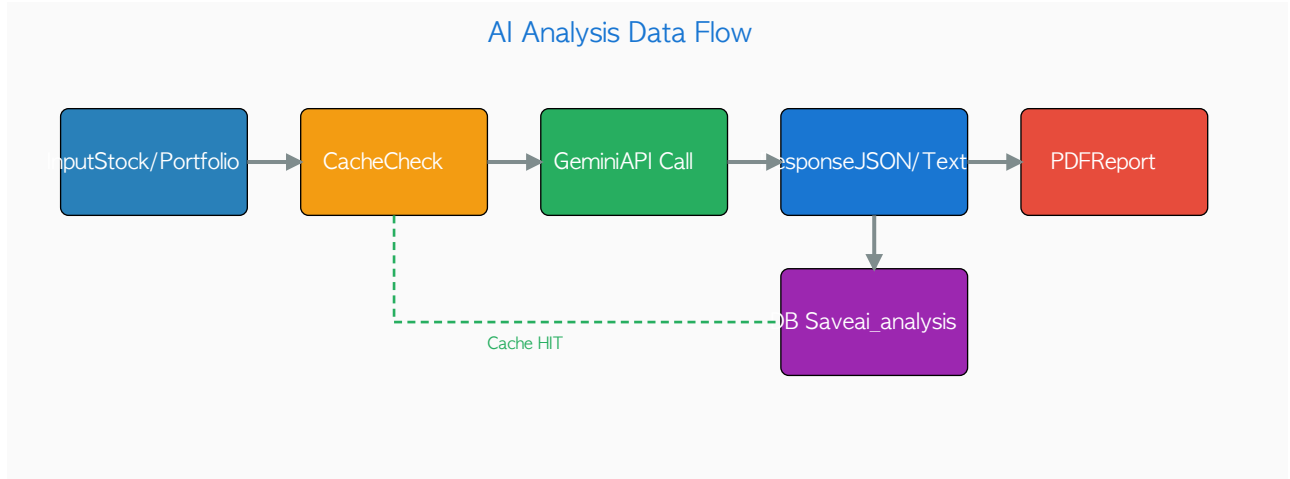
post_mortem	MODEL_PREMIUM	사후 분석 (성공/실패 원인)	12시간
trade_tracking	MODEL_SMART	매매 추적 분석	12시간
news_summary	MODEL_FAST	뉴스 요약	24시간
data_cleaning	MODEL_FAST	데이터 정제	-
fallback	MODEL_FAST	장애 시 대체	-

4. Fallback 전략 (Rate Limit 대응)



순서	모델	조건	대기 시간
1차 시도	gemini-3-pro-preview	요청 시 사용	-
2차 Fallback	gemini-1.5-pro	429/404 에러 시	2초
3차 Fallback	gemini-2.5-pro-preview	2차 실패 시	2초
최종 Fallback	gemini-2.0-flash	모든 실패 시	2초

5. AI 분석 데이터 흐름



6. ai_analysis 테이블 스키마 (database/models.py)

컬럼명	타입	설명	예시
id	Integer (PK)	자동 증가 ID	1, 2, 3...
stock_code	String(20)	종목코드 (FK → stocks)	015760, 006340
analysis_type	String(50)	분석 유형	STOCK, NEWS_THEME, PORTFOLIO
analysis_date	DateTime (TZ)	분석 일시 (UTC)	2025-12-03 01:25:56
model_name	String(100)	사용 모델	gemini-3-pro-preview
data_hash	String(64)	입력 데이터 SHA-256 해시	a1b2c3...
summary	Text	분석 요약	대원전선은 최근...
risk_factor	Text	리스크 요인	원자재 비용 상승...
action	String(20)	투자 의견	BUY, HOLD, SELL
confidence	Integer	신뢰도 (0-100)	85
raw_json	Text	전체 JSON 응답	{"fusion_score": 35, ...}
parent_id	Integer (FK)	부모 분석 ID	계층 구조용

7. 캐시 정책 (brain/ai_cache.py)

analysis_type	유효 시간	설명
STOCK	24시간	개별 종목 분석
PORTFOLIO	1시간	포트폴리오 분석 (실시간 반영 필요)
HOLDING	6시간	보유종목 상세 분석
NEWS_THEME	24시간	뉴스 테마 분석
NEW_STOCK	24시간	신규 종목 추천
KIS_DASHBOARD	1시간	KIS 대시보드
MIGRATION	6시간	이관 계획

daily_recommendation	12시간	일일 추천
DEFAULT	24시간	기본값

8. Analysis Type 분류

Type	설명	저장 위치	모델
STOCK	개별 종목 심층 분석	ai_analysis (stock_code=종목코드)	MODEL_PREMIUM
NEWS_THEME	뉴스 기반 테마 분석	ai_analysis (JSON in summary)	MODEL_FAST
PORTFOLIO	포트폴리오 전체 분석	ai_analysis (stock_code=KIS_PORTFOLIO_CODE)	MODEL_SMART
HOLDING	보유종목 분석	ai_analysis	MODEL_PREMIUM
CANDIDATE_PICK	매수 후보 종목	ai_analysis	MODEL_PREMIUM
CANDIDATE_MASTER	후보 종목 마스터 분석	ai_analysis	MODEL_PREMIUM
PROFIT_TAKING	익절 타이밍 분석	ai_analysis	MODEL_PREMIUM

9. raw_json 저장 구조 (STOCK 타입)

필드	타입	설명	필수
model	string	사용된 Gemini 모델	O
timestamp	string	분석 시간 YYYY-MM-DD HH:MM:SS	O
action	string	BUY/HOLD/SELL	O
overall_rating	string	STRONG BUY ~ STRONG SELL	O
fusion_score	int	Fusion Score (0-100)	O
confidence	int	확신도 (0-100)	O
risk_level	string	LOW/MEDIUM/HIGH	O
technical_score	int	기술적 분석 점수	-
supply_demand_score	int	수급 분석 점수	-
sentiment_score	int	심리 분석 점수	-
fundamental_score	int	펀더멘탈 점수	-
summary	string	분석 요약 (5-7문장 한국어)	O
key_points	array	핵심 포인트 리스트	O
action_items	array	권장 행동 리스트	-
risk_factors	array	리스크 요인 리스트	O

10. 파일별 AI 호출 현황

프로세스 파일	모델	용도
reporting/kis_dashboard.py	MODEL_FAST (기본)	KIS 포트폴리오 분석
reporting/stock_report.py	MODEL_PREMIUM	개별 종목 심층 분석
reporting/holding_report.py	MODEL_FAST	보유종목 피드백
reporting/generate_realtime_dashboard.py	MODEL_FAST	실시간 대시보드
reporting/new_stock_report.py	MODEL_SMART	신규종목 추천
fetchers/naver_finance/news.py	MODEL_FAST	뉴스 테마 분석

brain/strategy_architect.py	MODEL_PREMIUM	전략 설계
brain/integrated_analysis.py	MODEL_SMART	통합 분석

11. 모범 사례 및 규칙

권장 사항

- 기본 모델은 MODEL_FAST (gemini-2.0-flash) 사용 - 비용 효율적
- 심층 분석이 필요할 때만 MODEL_PREMIUM 사용
- DB 조회 시 항상 ORDER BY analysis_date DESC (최신순)
- raw_json에 전체 응답 저장하여 추후 활용 가능하게
- data_hash로 캐시 무효화 관리

금지 사항

- 거래 명령어를 AI에 직접 요청 (SafeExecutor 사용 필수)
- 불필요한 MODEL_PREMIUM 호출 (비용 증가)
- raw_json 없이 summary만 저장 (정보 손실)

12. API 호출 코드 예시

```
from brain.client import get_response_with_meta, MODEL_PREMIUM
from database.models import AIAnalysis
from database.connector import SessionLocal
import json

# 1. AI ■■■
result = get_response_with_meta(prompt, MODEL_PREMIUM)

# 2. JSON ■■
try:
    ai_data = json.loads(result['text'])
except:
    ai_data = {'summary': result['text']}

# 3. DB ■■
db = SessionLocal()
analysis = AIAnalysis(
    stock_code='006340',
    analysis_type='STOCK',
    model_name=result['model'],
    summary=ai_data.get('summary'),
    action=ai_data.get('action'),
    confidence=ai_data.get('confidence'),
    raw_json=result['text'] # ■■ ■■ (■■!)
)
db.add(analysis)
db.commit()
```

13. 관련 문서 참조

문서	내용	위치
Stocks_Logic.pdf	DB 테이블 전체 명세	reports/AEGIS/
Fetcher_Logic.pdf	Fetcher 자가 복구, Recovery Queue	reports/AEGIS/
Realtime_Logic.pdf	KB+KIS 실시간 대시보드	reports/AEGIS/

Realtime_Kis_Logic.pdf	KIS API 테이블 상세 명세	reports/AEGIS/
Picks_Logic.pdf	종목 선정 및 추천 로직	reports/AEGIS/