

gomgom-ai 프로젝트

본 문서는 gomgom-ai 프로젝트의 전체 아키텍처, API 흐름, 캐시 구조, 기능별 시퀀스, 데이터베이스 ERD 구조 및 기술 스택 특이사항까지 모두 포함한 정리본입니다.

1. API 흐름 개요

로그인 플로우 (SNS 로그인 연동)

- [프론트] SNS 로그인 버튼 클릭
- ↓
- [SNS] 인증 코드 발급 (카카오 / 네이버 / 구글)
- ↓
- [프론트] 인증 코드 → 백엔드 전송
- ↓
- [백엔드] 사용자 정보 요청 + JWT 토큰 발급
- ↓
- [Redis] 세션 저장
- ↓
- [프론트] JWT 토큰 저장 (localStorage 등)

- 사용 API: POST /auth/oauth/callback
- 사용 기술: OAuth2, JWT, Redis 세션
- 관련 파일: auth.py , core/oauth_client.py , core/session.py

감정 분석 → GPT 음식 추천

- [프론트] 감정 입력 (예: "우울해요")
- ↓
- [백엔드] GPT 프롬프트 생성
- ↓
- [GPT] 음식 키워드 추천

- ↓
- [Redis] 추천 결과 캐싱
- ↓
- [프론트] 추천 리스트 표시

- 사용 API: POST /recommendations/emotion
- 관련 파일: recommend.py , services/gpt_service.py , core/cache.py

음식점 검색 (요기요 API 활용)

[프론트] 음식 키워드 선택 (예: "족발")

↓

[백엔드] 요기요 API 요청

↓

[백엔드] 음식점 필터링 및 정렬

↓

[Redis] 음식점 결과 캐싱

↓

[프론트] 음식점 리스트 표시

- 사용 API: GET /store/search
- 관련 파일: services/yogiyo_crawler.py , utils/extract_keywords.py

레시피 및 재료 정보 출력

[프론트] 음식 클릭 (예: "김치찌개")

↓

[백엔드] Spoonacular API 호출

↓

[백엔드] 재료 번역 (영어 → 한글)

↓

[프론트] 조리법, 재료, 이미지 표시

- 사용 API: GET /recipe
- 관련 파일: services/recipe_service.py , utils/ingredient_translator.py

2. Redis 캐시 구조 요약

상황 캐시 키 내용 TTL
감정 분석 결과 emotion:{입력값} GPT 음식 키워드 리스트 10분
음식점 리스트 store:{음식명} 요기요 음식점 JSON 결과 10분
사용자 세션 session:{user_id} 로그인 사용자 정보, JWT 1일

3. 전체 기능별

목적: 사용자 인증 및 JWT 발급

- [프론트] 사용자가 SNS 로그인 버튼 클릭
- [SNS 서버] 인증 코드 발급 후 Redirect URI 호출
- [프론트] 인증 코드 수신 후 백엔드에 전달 (/auth/oauth/callback)
- [백엔드] SNS API를 통해 사용자 정보 조회
- [백엔드] 내부 사용자 DB에 등록 여부 확인
- [백엔드] JWT 토큰 생성 (액세스, 리프레시)
- [Redis] 사용자 세션 저장
- [프론트] JWT 토큰 수신 및 로컬 저장 (localStorage)

2. 감정 입력 → GPT 추천 키워드

목적: 감정 또는 상황에 따라 GPT가 어울리는 음식 키워드 추천

- [프론트] 사용자 감정 입력 (예: 우울할 때 먹고 싶은 음식)
- [프론트] /recommendations/emotion API 호출
- [백엔드] 감정 텍스트를 GPT 프롬프트로 가공
- [GPT API] 감정에 어울리는 음식 키워드 응답
- [백엔드] 추천 결과 Redis 캐시 저장
- [프론트] 추천 음식 리스트 표시

3. 음식 키워드 선택 → 요기요 음식점 검색

목적: 사용자가 선택한 음식 키워드로 실제 음식점 검색

검색

- [프론트] 음식 키워드 클릭 (예: "피자")
- [프론트] /restaurants/search?keyword=피자 호출
- [백엔드] 요기요 비공식 API 또는 크롤러로 음식점 정보 수집
- [백엔드] 평점, 리뷰 등 기준으로 정렬/필터링
- [Redis] 음식점 리스트 캐시 저장
- [프론트] 음식점 리스트 및 썸네일 표시

4. 음식 클릭 → 레시피 및 재료 정보

목적: 사용자가 선택한 음식에 대한 레시피

제공

1. [프론트] 음식점 또는 음식 클릭 → 상세 요청
2. [프론트] /recipes?food_name=김치찌개
호출 3.
[백엔드] Spoonacular API 또는 내부 DB에서 레시피
- 검색 4.
[백엔드] 영어 재료명 → 한글 번역
5. [백엔드] 재료 + 조리법 구성 후 반환

3

6. [프론트] 레시피 화면에 출력

5. 추천 결과 재사용 (캐시 기반)

목적: 동일 감정/키워드 요청 시 캐시 활용으로 속도 향상

1. [프론트] 이전에 입력한 감정 또는 키워드로 다시 요청
2. [프론트] /recommend_result/cached?key=emotion:우울해요
호출 3.
[백엔드] Redis에서 캐시된 추천 결과 탐색
4. [프론트] 캐시된 결과 즉시 렌더링

6. 사용자 위치 기반 검색 (선택 기능)

목적: 사용자의 현재 위치 근처 음식점 또는 카테고리 기반 탐색

1. [프론트] 사용자 위치 정보 수집 (latitude, longitude) 2.
[프론트] /locations/nearby?lat=...&lng=...
호출 3.
[백엔드] 위치 기반 검색 알고리즘 실행
4. [백엔드] 주변 음식점 또는 인기 메뉴 추천
5. [프론트] 결과 표시

4. ERD 구조 요약

주요 테이블 목록 및 설명

테이블명 설명 주요 컬럼 관계		
users	<p>사용자 정보 저 장</p> <p>id, email, full_name, is_active</p>	1:N ↔ recommendations
auth_user	<p>Django 기본 사 용자 모 델</p> <p>username, email, is_staff 관리용</p>	
gomgom_ai_recommendation	<p>감정 기 반 추천 결과</p> <p>input_text, recommended_store_chars, matched_restaurant_id</p>	↔ users (user_id FK)
recommendations	<p>음식 추</p> <p>천 기록 user_id, recipe_id, score</p>	↔ users, recipes
recipes	<p>레시피 상세 정 보</p> <p>name, instructions, difficulty, servings</p>	1:N ↔ recommendations
ingredients	<p>재료 기 본 정보</p> <p>name, price, unit, category</p>	1:N ↔ ingredient_ko

4

테이블명 설명 주요 컬럼 관계		
ingredient_ko	<p>재료 한 글 번역 정보</p> <p>eng_name, kor_name</p>	↑ gomgom_ai_ingredient

	추천 결	
gomgom_ai_ingredient	과 연동 재료	name, price, ko_info_id ↔ ingredient_ko
gomgom_ai_recipe	GPT 기 반 추천 용 레시 피	title, description, category, keywords 외부 레시피 연계
locations	위치 기 반 음식 점 정보	name, latitude, longitude, category 위치 기반 추천 가능
auth_group, auth_permission	Django 내장 권 한 시스 템	- 관리 목적
django_session	세션 관	session_key, session_data Redis 또는 DB 기반
django_migrations	마이그 레이션 이력	app, name, applied Alembic 유사
django_content_type	Django 앱 모델 정보	app_label, model admin UI 연계

핵심 관계 구조 (요약)

```
[users]
└─< gomgom_ai_recommendation
└─< gomgom_ai_recipe
└─< gomgom_ai_ingredient
└─< ingredient_ko

[recipes]
└─< recommendations
└─< users
```

5. 기술 스택 기반 특이사항

- **FastAPI** → 사용자 정의 모델은 gomgom_ai_ 접두어 사용

5

- **Django** → auth_user , auth_group , session 등 admin 인증 구조에 사용 •
Redis → 세션 저장소 및 캐시 레이어로 동시 활용됨
- 다국어 재료 처리 → ingredient_ko , kor_name , eng_name 구조 활용

6. 활용 예시

- **Emotion → Recipe Recommendation:** gomgom_ai_recommendation.input_text → gomgom_ai_recipe or recipes
- 식재료 가격 비교: gomgom_ai_ingredient → price , unit •
추천 기록 통계화: recommendations.score
- 위치 기반 **UX** 구현: locations 테이블의 위경도 기반 정렬

작성자: 민아 + ANIM\ 버전: 1.0\ 최종 수정일: 2025-06-18

