

Last Mile

API Specification v1.0

TEAM 모세의 기적

초정밀 보행 최단 경로 최적화

Base URL : https://github.com/caramel-Case/lastmile/tree/Lastmile_V-2.1.0

1. 개요 (Overview)

Last Mile은 사용자의 성향 및 상황을 반영한 최적의 보행 경로를 제공하는 웹 어플리케이션입니다. 단순한 최단 거리가 아닌, 경사로·계단·조명 밀도·CCTV 등 다양한 요소를 고려하여 사용자 맞춤형 경로를 산출합니다.

1.1 기술 스택

Backend: Go (goroutine 기반 병렬 가중치 연산) , 네이버 API

Database: PostgreSQL + PostGIS (공간 데이터 처리)

Frontend: React + react-naver-maps

Data Source: OSM, 공공데이터 포털

1.2 인증 방식

모든 인증이 필요한 요청에는 Authorization 헤더에 Bearer Token을 포함해야 합니다.

Authorization: Bearer {access_token}

1.3 공통 응답 형식

모든 응답은 JSON 형식이며, 성공 시 status 필드에 "success", 실패 시 "error"가 반환됩니다.

1.4 공통 에러 코드

Status	Description
400	Bad Request – 요청 파라미터 오류
401	Unauthorized – 인증 토큰 누락 또는 만료
403	Forbidden – 권한 부족
404	Not Found – 리소스 존재하지 않음

429	Too Many Requests – 요청 제한 초과
500	Internal Server Error – 서버 내부 오류

2. Users Domain

사용자 계정 관리, 인증, 보행 선호도 설정을 담당하는 도메인입니다.

POST /api/v1/users/signup

계정 생성 및 보행 환경 기본값 설정. 회원가입 시 모든 가중치는 0으로 초기화됩니다.

Request Body

Parameter	Type	Required	Description
email	string	필수	사용자 이메일 주소
password	string	필수	비밀번호 (8자 이상, 복잡성 검증)
user_type	string	필수	standard wheelchair stroller

Response (201 Created)

Field	Type	Description
user_id	string (UUID)	생성된 사용자 고유 ID
email	string	등록된 이메일
user_type	string	사용자 유형
preferences	object	초기화된 가중치 설정 (모두 0)

보안 사항

비밀번호 복잡성 검증 (8자 이상, 대소문자 + 숫자 + 특수문자)

중복 가입 방지 (409 Conflict)

POST /api/v1/users/login

이메일/비밀번호 인증 후 토큰 발급

Request Body

Parameter	Type	Required	Description
email	string	필수	사용자 이메일
password	string	필수	비밀번호

Response (200 OK)

Field	Type	Description
access_token	string	JWT Access Token
refresh_token	string	JWT Refresh Token
expires_at	string (ISO 8601)	Access Token 만료 시각

보안 사항

Timing Attack 방어 로직 적용 (constant-time comparison)

POST /api/v1/users/refresh

Refresh Token을 사용하여 새로운 Access Token 쌍 발급

Request Body

Parameter	Type	Required	Description
refresh_token	string	필수	기존 발급받은 Refresh Token

Response (200 OK)

Field	Type	Description
access_token	string	새로운 Access Token
refresh_token	string	새로운 Refresh Token
expires_at	string (ISO 8601)	만료 시각

보안 사항

Token Rotation 적용: 기존 Refresh Token은 즉시 무효화

GET /api/v1/users/me

현재 로그인한 사용자의 프로필 및 가중치 설정 조회

Headers

Parameter	Type	Required	Description
Authorization	string	필수	Bearer {access_token}

Response (200 OK)

Field	Type	Description
user_id	string (UUID)	사용자 ID
email	string	이메일
user_type	string	사용자 유형
preferences	object	가중치 설정 객체
preferences.stair_avoidance	integer (0-3)	계단 회피 강도
preferences.slope_avoidance	integer (0-3)	경사로 회피 강도
preferences.safety_preference	integer (0-3)	안심 경로 선호도
preferences.main_road_preference	integer (0-3)	대로 우선 강도

PATCH /api/v1/users/me/preferences

사용자의 상세 회피/선호 강도(0-3) 수정. 부분 업데이트 지원.

Request Body (부분 업데이트)

Parameter	Type	Required	Description
stair_avoidance	integer (0-3)	선택	0=off, 1=하, 2=중, 3=상
slope_avoidance	integer (0-3)	선택	0=off, 1=하, 2=중, 3=상
safety_preference	integer (0-3)	선택	안심 경로 선호도

main_road_preference	integer (0-3)	선택	대로 우선 강도
----------------------	---------------	----	----------

Response (200 OK)

수정된 preferences 객체 반환

DELETE /api/v1/users/me

회원 탈퇴 및 PII(개인식별정보) 즉시 익명화 처리

보안 사항

Soft Delete 적용 (데이터 복구 기간 운영)

모든 Refresh Token 즉시 파기

Response (204 No Content)

GET /api/v1/spatial/gateways/pending

[관리자 전용] 검토 대기 중인 지름길 제보 목록 조회. Role: admin 권한 필요.

Response (200 OK)

Field	Type	Description
gateways	array	대기 중인 지름길 목록
gateways[].id	string (UUID)	지름길 ID
gateways[].lat	float	위도
gateways[].lng	float	경도
gateways[].type	string	유형 (side_door, underpass 등)
gateways[].status	string	pending
gateways[].reported_by	string (UUID)	제보자 ID

3. Spatial Domain

지도 데이터, 공간 정보, 사용자 제보를 담당하는 도메인입니다. PostGIS 기반 공간 쿼리를 활용합니다.

POST /api/v1/spatial/gateways

기존 지도에 없는 건물 쪽문, 지하도 출입구, 공원 진입 통로 등 등록. 초기 상태는 *pending*으로 저장되며, 관리자 승인 후 경로 엔진에 반영.

Request Body

Parameter	Type	Required	Description
lat	float	필수	위도 좌표
lng	float	필수	경도 좌표
type	string	필수	side_door underpass park_entry shortcut
properties	object	선택	추가 메타데이터 (description, photo_url 등)

Response (201 Created)

Field	Type	Description
gateway_id	string (UUID)	생성된 지름길 ID
status	string	pending (관리자 승인 대기)

POST /api/v1/spatial/reports

공사, 사고, 가로등 고장 등 실시간 위험 요소 제보

Request Body

Parameter	Type	Required	Description
lat	float	필수	위험 요소 위도
lng	float	필수	위험 요소 경도
type	string	필수	construction accident broken_light other

description	string	선택	상세 설명
expires_at	string (ISO 8601)	선택	제보 만료 시각 (기본: 24시간 후)

Response (201 Created)

Field	Type	Description
report_id	string (UUID)	제보 ID
status	string	active

GET /api/v1/spatial/areas/{area_id}/visibility-graph

광장이나 공원 내부를 가로지르는 최적 가시 그래프 경로 계산

Path Parameters

Parameter	Type	Required	Description
area_id	string (UUID)	필수	공원/광장 영역 ID

Response (200 OK)

Field	Type	Description
nodes	array	가시 그래프 노드 목록
edges	array	노드 간 가시 연결 정보
entry_points	array	공원 진입점 좌표
exit_points	array	공원 진출점 좌표

GET /api/v1/spatial/areas/{area_id}/geometry

공원의 외곽 경계, 장애물(Holes), 출입구(Gateways) 좌표 데이터 제공

Response (200 OK)

Field	Type	Description
boundary	GeoJSON Polygon	공원 외곽 경계
holes	array of Polygon	내부 장애물 영역

gateways	array of Point	출입구 좌표 목록
----------	----------------	-----------

GET /api/v1/spatial/search*네이버 API를 활용한 장소 검색 및 좌표 반환***Query Parameters**

Parameter	Type	Required	Description
query	string	필수	검색어 (예: "서울역")
limit	integer	선택	결과 수 제한 (기본: 5)

Response (200 OK)

Field	Type	Description
results	array	검색 결과 목록
results[].name	string	장소명
results[].lat	float	위도
results[].lng	float	경도
results[].address	string	주소

GET /api/v1/spatial/graph-context*특정 영역(BBox) 내의 모든 노드와 엣지 데이터를 해시 테이블 구성용 JSON으로 반환***Query Parameters**

Parameter	Type	Required	Description
min_lat	float	필수	최소 위도
min_lng	float	필수	최소 경도
max_lat	float	필수	최대 위도
max_lng	float	필수	최대 경도

Response (200 OK)

Field	Type	Description
nodes	array	영역 내 노드 목록 (id, lat, lng, tags)
edges	array	영역 내 엣지 목록 (from, to, distance, features)
areas	array	영역 내 공원/광장 polygon 데이터

4. Navigation Domain

핵심 경로 엔진을 담당하는 도메인입니다. 다익스트라 / A* 알고리즘 기반으로 사용자 맞춤형 가중치 경로를 산출합니다.

POST /api/v1/navigation/route

[핵심 API] 출발/도착지와 사용자 강도 설정을 조합하여 3종 경로(최단, 균형, 안심) 산출

Request Body

Parameter	Type	Required	Description
origin	object	필수	출발지 {lat: float, lng: float}
destination	object	필수	도착지 {lat: float, lng: float}
custom_weights	object	선택	사용자 가중치 설정
custom_weights.stair	integer (0-3)	선택	계단 회피 강도
custom_weights.slope	integer (0-3)	선택	경사로 회피 강도
custom_weights.safety	integer (0-3)	선택	안심 경로 선호도
custom_weights.main_road	integer (0-3)	선택	대로 우선 강도

처리 로직

1. 출발/도착지 기반 BBox(Bounding Box) 설정
2. Spatial 도메인으로부터 해당 영역의 그래프 데이터를 메모리 해시 테이블로 로드
3. 사용자의 상/중/하 가중치를 해시 테이블의 각 Edge에 즉시 적용
4. 공원 내부 진입 시 출입구 간 최단 가시 거리 계산

5. 길찾기 엔진 가동 (다익스트라 / A* 알고리즘)

Cost Function

$$\text{Total_Cost} = \text{Distance} \times (1 + \sum(\text{Feature_Penalty} \times \text{User_Intensity}))$$

예시: 계단 회피(상) 선택 시 계단 포함 간선의 가중치를 5배로 상향 조정

Response (200 OK)

Field	Type	Description
routes	array	3종 경로 배열
routes[].type	string	shortest balanced safest
routes[].geometry	GeoJSON LineString	경로 좌표 배열
routes[].distance_m	float	총 거리 (m)
routes[].duration_min	float	예상 소요 시간 (분)
routes[].safety_score	float (0-100)	경로 안전 점수
routes[].detour_index	float	우회 지수 (직선 대비 비율)
routes[].segments	array	구간별 상세 정보

GET /api/v1/spatial/amenities

현재 지도 뷰포트 내의 CCTV, 가로등, 안전벨 등 POI 데이터 조회

Query Parameters

Parameter	Type	Required	Description
min_lat	float	필수	뷰포트 최소 위도
min_lng	float	필수	뷰포트 최소 경도
max_lat	float	필수	뷰포트 최대 위도
max_lng	float	필수	뷰포트 최대 경도
types	string	선택	cctv,streetlight,safety_bell (콤마 구분)

Response (200 OK)

Field	Type	Description
amenities	array	POI 목록
amenities[].type	string	cctv streetlight safety_bell
amenities[].lat	float	위도
amenities[].lng	float	경도
amenities[].source	string	데이터 출처

GET /api/v1/navigation/edges/{edge_id}/safety-score

특정 길(Edge)의 조명 밀도, 사고 이력 기반 통합 안전 점수 반환

Path Parameters

Parameter	Type	Required	Description
edge_id	string	필수	Edge 고유 ID

Response (200 OK)

Field	Type	Description
edge_id	string	Edge ID
safety_score	float (0-100)	통합 안전 점수
light_density	float	조명 밀도 (ea/100m)
cctv_count	integer	근처 CCTV 수
accident_history	integer	사고 이력 건수

GET /api/v1/navigation/detour-index

추천 경로가 직선 거리 대비 얼마나 우회하는지 분석 (성능 지표)

Query Parameters

Parameter	Type	Required	Description
route_id	string	필수	분석 대상 경로 ID

Response (200 OK)

Field	Type	Description
route_id	string	경로 ID
straight_distance_m	float	직선 거리 (m)
route_distance_m	float	실제 경로 거리 (m)
detour_index	float	우회 지수 (경로/직선)

5. ETL Domain

데이터 파이프라인, 배치 처리, 가중치 피드백 루프를 담당하는 내부 도메인입니다.

PUT /api/v1/internal/etl/safety-data/{region_code}

[내부용] 공공데이터 포털의 안전 시설물 데이터를 Upsert 방식으로 DB 동기화

Path Parameters

Parameter	Type	Required	Description
region_code	string	필수	지역 코드 (예: 11680 → 강남구)

Request Body

Parameter	Type	Required	Description
data_type	string	필수	cctv streetlight safety_bell
items	array	필수	시설물 목록 [{lat, lng, properties}]

Response (200 OK)

Field	Type	Description
inserted	integer	새로 삽입된 건수
updated	integer	업데이트된 건수

region_code	string	대상 지역 코드
-------------	--------	----------

POST /api/v1/spatial/gpx/upload

실제 보행 GPX 데이터를 분석하여 지도에 없는 새로운 길 후보지 추출

Request Body

Parameter	Type	Required	Description
file	binary (GPX)	필수	GPX 파일 (multipart/form-data)

Response (200 OK)

Field	Type	Description
new_paths	array	발견된 새로운 길 후보 목록
new_paths[].geometry	GeoJSON LineString	경로 좌표
new_paths[].confidence	float (0-1)	신뢰도 점수

POST /api/v1/navigation/routes/{route_id}/feedback

[중요] 사용자의 경로 만족도 피드백을 수집하여 해당 구역 가중치 보정 로직 트리거

Path Parameters

Parameter	Type	Required	Description
route_id	string	필수	경로 ID

Request Body

Parameter	Type	Required	Description
rating	integer (1-5)	필수	만족도 평점
issues	array of string	선택	steep_slope dark_path construction etc
comment	string	선택	자유 의견

Response (201 Created)

Field	Type	Description
feedback_id	string (UUID)	피드백 ID
weight_update_triggered	boolean	가중치 보정 트리거 여부

GET /api/v1/internal/etl/logs

[내부용] 배치 작업의 성공/실패 여부 및 데이터 업데이트 로그 확인

Query Parameters

Parameter	Type	Required	Description
page	integer	선택	페이지 번호 (기본: 1)
limit	integer	선택	페이지당 건수 (기본: 20)
status	string	선택	success failed running

Response (200 OK)

Field	Type	Description
logs	array	ETL 로그 목록
logs[].job_id	string	작업 ID
logs[].type	string	작업 유형 (safety_sync, osm_update 등)
logs[].status	string	success failed running
logs[].started_at	string (ISO 8601)	시작 시각
logs[].completed_at	string (ISO 8601)	완료 시각
logs[].records_affected	integer	처리된 레코드 수
total	integer	전체 로그 수

6. 데이터 흐름 (Data Flow)

6.1 경로 탐색 플로우

1. 프론트엔드: 사용자가 목적지 텍스트 입력
2. GET /api/v1/spatial/search 호출 → 좌표 획득
3. POST /api/v1/navigation/route 호출 → 경로(GeoJSON) 수신
4. 수신된 좌표 배열을 react-naver-maps에서 Polyline으로 렌더링
5. 사용자 룡탭 → POST /api/v1/spatial/gateways로 지름길 제보

6.2 경로 계산 엔진 상세

계산 시점: 사용자가 '경로 찾기' 버튼을 누르는 순간

계산 장소: navigation/engine.go (메모리 해시 테이블 + goroutine)

공원 처리: polygon 데이터로 취급, 진입점-진출점 간 장애물이 없으면 직선 거리(Euclidean Distance)로 가중치 계산

6.3 데이터 검증 규칙

OSM 데이터 수집 시 보행 가능 도로(highway) 태그가 명확한 데이터만 필터링

공공데이터(CCTV 등)와 위치가 중복되는 경우 교차 검증하여 신뢰도 점수 부여