Interest Map MVP API Specification (v3)

Query policy: profile/gallery 조회는 GET /me 로 통일

1. Common Rules

Base URL (Emulator)	http://10.0.2.2:8000
Base URL (Local)	http://127.0.0.1:8000
Auth Header	Authorization: Bearer <access_token>
Content-Type	application/json
Date Format	ISO-8601 (e.g., 2026-01-09T11:10:00+09:00)

Common Error Response (JSON)

```
{
  "error": {
    "code": "ERROR_CODE",
    "message": "Human readable message",
    "details": { "optional": "any" }
  }
}
```

2. Auth

POST /auth/kakao

Validate Kakao access token, upsert user, and issue JWT access token.

Request (JSON)

```json
{
  "access_token": "kakao_access_token_string"
}
```

Response 200 (JSON)

```json
{
  "access_token": "jwt_access_token_string",
  "token_type": "bearer",
  "user": {
    "id": "6c7f0f9e-9ed8-4c7a-8fd2-0fd2cf40d3a2",
    "nickname": "승운",
    "profile_image_url": "https://k.kakaocdn.net/.../profile.jpg",
    "primary_photo_url": null
  }
}
```

Response 401 (JSON) - invalid Kakao token

```json
{
  "error": {
    "code": "KAKAO_TOKEN_INVALID",
    "message": "카카오 토큰이 유효하지 않습니다."
  }
}
```

3. Me (Profile + Gallery)

GET /me

Return my profile and interest data. Includes gallery photos to avoid extra fetch.

Response 200 (JSON)

```json
{
  "id": "6c7f0f9e-9ed8-4c7a-8fd2-0fd2cf40d3a2",
  "nickname": "승운",
  "profile_image_url": "https://k.kakaocdn.net/.../profile.jpg",
  "primary_photo_url": "https://cdn.example.com/photos/abc.jpg",
  "profile_data": {
    "bio": "AI/영화/위스키 좋아해요",
    "interests": ["영화", "게임", "AI"],
    "keywords": ["서스펜스", "미감", "플롯 밀도"],
    "likes": {
      "movie": ["SF", "thriller"],
      "music": ["jazz"]
    }
  },
  "photos": [
    {
      "id": "2d6e9be6-72e8-4b5b-9f55-7b1ee7b79a3d",
      "url": "https://cdn.example.com/photos/1.jpg",
      "sort_order": 10,
      "is_primary": true,
      "created_at": "2026-01-09T10:00:00+09:00"
    },
    {
      "id": "3bb6c8c2-c50a-4c69-86f9-4c2b9e3c173c",
      "url": "https://cdn.example.com/photos/2.jpg",
      "sort_order": 20,
      "is_primary": false,
      "created_at": "2026-01-09T10:05:00+09:00"
    }
  ],
  "embedding": {
    "status": "ready",
    "model": "text-embedding-3-large",
    "updated_at": "2026-01-09T11:10:00+09:00"
  }
}
```

PATCH /me

Update nickname and/or profile_data (partial update).

Request (JSON)

```json
{
  "nickname": "승운",
  "profile_data": {
    "bio": "서스펜스랑 미장센 좋아함",
    "interests": ["영화", "전시", "AI"],
    "keywords": ["서스펜스", "미감", "밀도"]
  }
}
```

Response 200 (JSON)

```json
{
  "ok": true
}
```

4. Photos (Actions)

POST /me/photos

Register an uploaded image URL. Server assigns sort_order at the end of the list.

Request (JSON)

```
{
  "url": "https://cdn.example.com/photos/new.jpg",
  "make_primary": false
}
```

Response 201 (JSON)

```
{
  "id": "9b2a2b4e-4c90-4e6e-9d8e-0a1e1b2c3d4e",
  "url": "https://cdn.example.com/photos/new.jpg",
  "sort_order": 30,
  "is_primary": false,
  "created_at": "2026-01-09T11:15:00+09:00"
}
```

Response 409 (JSON) - duplicate URL for same user

```
{
  "error": {
    "code": "PHOTO_URL_DUPLICATE",
    "message": "이미 등록된 이미지 URL입니다."
  }
}
```

PATCH /me/photos/order

Update photo ordering (e.g., drag-and-drop result).

Request (JSON)

```
{
  "orders": [
    { "id": "3bb6c8c2-c50a-4c69-86f9-4c2b9e3c173c", "sort_order": 10 },
    { "id": "2d6e9be6-72e8-4b5b-9f55-7b1ee7b79a3d", "sort_order": 20 }
  ]
}
```

Response 200 (JSON)

```
{ "ok": true }
```

POST /me/photos/{photo_id}/primary

Set a photo as primary (DB enforces only one primary per user).

Response 200 (JSON)

```
{
  "ok": true,
  "primary_photo_url": "https://cdn.example.com/photos/2.jpg"
}
```

DELETE /me/photos/{photo_id}

Delete a photo from my gallery.

Response 200 (JSON)

```
{ "ok": true }
```

5. Groups

GET /groups

List available groups (seeded rooms supported).

Response 200 (JSON)

```json
{
  "items": [
    {
      "id": "b3b8f5fd-51fd-4d5e-93d7-3fb2f7e8f001",
      "name": "SF 영화방",
      "description": "SF 좋아하는 사람들",
      "member_count": 12,
      "is_member": false
    },
    {
      "id": "5c4a2d1e-6b11-4d4f-a5b1-9f7c2d3e4f55",
      "name": "전시/미감 방",
      "description": "미장센/미감 위주 취향",
      "member_count": 7,
      "is_member": true
    }
  ]
}
```

POST /groups/{group_id}/join

Join a group.

Response 200 (JSON)

```json
{ "ok": true }
```

GET /groups/{group_id}/members

List members in a group.

Response 200 (JSON)

```json
{
  "items": [
    {
      "user_id": "6c7f0f9e-9ed8-4c7a-8fd2-0fd2cf40d3a2",
      "nickname": "승운",
      "primary_photo_url": "https://cdn.example.com/photos/abc.jpg"
    },
    {
      "user_id": "c1d2e3f4-1111-2222-3333-444455556666",
      "nickname": "지민",
      "primary_photo_url": null
    }
  ]
}
```

6. Group Interest Map

GET /groups/{group_id}/interest-map

Return 2D coordinates for all group members to render an interest map on the group screen.

Response 200 (JSON)

```json
{
  "group": {
    "id": "b3b8f5fd-51fd-4d5e-93d7-3fb2f7e8f001",
    "name": "SF 영화방"
  },
  "layout": {
    "method": "umap",
    "version": "v1",
    "generated_at": "2026-01-09T12:10:00+09:00"
  },
  "nodes": [
    {
      "user_id": "6c7f0f9e-9ed8-4c7a-8fd2-0fd2cf40d3a2",
      "nickname": "승운",
      "primary_photo_url": "https://cdn.example.com/photos/abc.jpg",
      "x": 0.12,
      "y": -0.48,
      "embedding_status": "ready"
    },
    {
      "user_id": "c1d2e3f4-1111-2222-3333-444455556666",
      "nickname": "지민",
      "primary_photo_url": null,
      "x": -0.33,
      "y": 0.22,
      "embedding_status": "missing"
    }
  ],
  "edges": [
    {
      "from_user_id": "6c7f0f9e-9ed8-4c7a-8fd2-0fd2cf40d3a2",
      "to_user_id": "c1d2e3f4-1111-2222-3333-444455556666",
      "similarity": 0.71
    }
  ]
}
```

7. Group Chat (REST)

GET /groups/{group_id}/messages

Fetch chat history. Use limit and before for pagination (infinite scroll).

Example URL

```
/groups/{group_id}/messages?limit=30&before=2026-01-09T12:00:00+09:00
```

Response 200 (JSON)

```
{
  "items": [
    {
      "id": "8f1a0b2c-3333-4444-5555-666677778888",
      "group_id": "b3b8f5fd-51fd-4d5e-93d7-3fb2f7e8f001",
      "sender": {
        "user_id": "6c7f0f9e-9ed8-4c7a-8fd2-0fd2cf40d3a2",
        "nickname": "승운",
        "primary_photo_url": "https://cdn.example.com/photos/abc.jpg"
      },
      "content": { "text": "안녕하세요!" },
      "created_at": "2026-01-09T12:05:10+09:00"
    }
  ],
  "next_before": "2026-01-09T11:58:00+09:00"
}
```

POST /groups/{group_id}/messages

Send a chat message (REST).

Request (JSON)

```
{
  "text": "안녕하세요!"
}
```

Response 201 (JSON)

```
{
  "id": "8f1a0b2c-3333-4444-5555-666677778888",
  "group_id": "b3b8f5fd-51fd-4d5e-93d7-3fb2f7e8f001",
  "sender": {
    "user_id": "6c7f0f9e-9ed8-4c7a-8fd2-0fd2cf40d3a2",
    "nickname": "승운",
    "primary_photo_url": "https://cdn.example.com/photos/abc.jpg"
  },
  "content": { "text": "안녕하세요!" },
  "created_at": "2026-01-09T12:05:10+09:00"
}
```

8. Embedding (Optional for MVP)

POST /me/embedding/rebuild

Rebuild interest embedding from profile_data and store into user_embeddings.

Response 200 (JSON)

```
{
  "ok": true,
  "embedding": {
    "status": "ready",
    "model": "text-embedding-3-large",
    "updated_at": "2026-01-09T11:20:00+09:00"
  }
}
```