

# Nugulmap API Specification

---

This document describes the API endpoints for the Nugulmap backend service.

## Base URL

The base URL for all API endpoints is typically `http://localhost:8080` (or your deployed server address).

## Response Format

Most endpoints return a JSON response with the following structure:

```
{
  "success": boolean,
  "message": "string",
  "data": { ... }
}
```

---

## 1. Users (`/users`)

### Get User

- **URL:** `/users/{id}`
- **Method:** `GET`
- **Description:** Retrieves user details by ID.
- **Parameters:**
  - `id` (Path, Long): User ID.
- **Response:**

```
{
  "id": 1,
  "email": "user@example.com",
  "nickname": "Nickname",
  "profileImage": "/images/filename.jpg",
  "createdAt": "2023-..."
}
```

### Create User

- **URL:** `/users`
- **Method:** `POST`
- **Content-Type:** `multipart/form-data`
- **Description:** Creates a new user. Handles OAuth user creation with optional profile image.

- **Parameters:**

- **userData** (Part, JSON String): JSON string representing **UserRequest**.

```
{
  "email": "user@example.com",
  "oauthId": "12345",
  "oauthProvider": "kakao|google|naver",
  "nickname": "Nickname"
}
```

- **profileImage** (Part, File, Optional): Profile image file.

- **Response:** JSON with created user details.

## Update User

- **URL:** `/users/{id}`
- **Method:** **PUT**
- **Content-Type:** `multipart/form-data`
- **Description:** Updates user information and/or profile image.
- **Parameters:**
  - **id** (Path, Long): User ID.
  - **userData** (Part, JSON String): JSON string representing **UserRequest**.
  - **profileImage** (Part, File, Optional): New profile image file.
- **Response:** JSON with updated user details.

## Update Profile Image Only

- **URL:** `/users/{id}/profile-image`
- **Method:** **PUT**
- **Content-Type:** `multipart/form-data`
- **Description:** Updates only the user's profile image.
- **Parameters:**
  - **id** (Path, Long): User ID.
  - **profileImage** (Part, File, Required): New profile image file.
- **Response:** JSON with new profile image filename.

## Delete User

- **URL:** `/users/{id}`
- **Method:** **DELETE**
- **Description:** Deletes a user.
- **Parameters:**
  - **id** (Path, Long): User ID.
- **Response:** Success message.

---

## 2. Zones (`/zones`)

## Create Zone

- **URL:** `/zones`
- **Method:** `POST`
- **Content-Type:** `multipart/form-data`
- **Description:** Creates a new smoking zone.
- **Parameters:**
  - **data** (Part, JSON String): JSON string representing `ZoneRequest`.

```
{
  "region": "Seoul",
  "type": "Outdoor",
  "subtype": "Booth",
  "description": "Description...",
  "latitude": 37.5665,
  "longitude": 126.9780,
  "size": "Medium",
  "address": "Address...",
  "user": "UserId or Name"
}
```

- **image** (Part, File, Optional): Zone image file.
- **Response:** JSON with created zone details.

## Get Zone

- **URL:** `/zones/{id}`
- **Method:** `GET`
- **Description:** Retrieves zone details by ID.
- **Parameters:**
  - **id** (Path, Integer): Zone ID.
- **Response:**

```
{
  "success": true,
  "data": {
    "zone": {
      "id": 1,
      "region": "Seoul",
      ...
      "image": "filename.jpg"
    }
  }
}
```

## Get All Zones

- **URL:** `/zones`
- **Method:** `GET`
- **Description:** Retrieves all zones.
- **Response:** JSON list of zones.

### Get All Zones (Paged)

- **URL:** `/zones/paged`
- **Method:** `GET`
- **Description:** Retrieves zones with pagination.
- **Parameters:**
  - **page** (Query, int): Page number (0-indexed).
  - **size** (Query, int): Page size (default 20).
  - **sort** (Query, string): Sort field (default "id").
- **Response:** JSON with zones list and pagination info.

### Search Zones by Radius

- **URL:** `/zones`
- **Method:** `GET`
- **Description:** Retrieves zones within a specific radius of a coordinate.
- **Parameters:**
  - **latitude** (Query, double): Center latitude.
  - **longitude** (Query, double): Center longitude.
  - **radius** (Query, int): Radius in meters.
- **Response:** JSON list of zones within the radius.

### Update Zone

- **URL:** `/zones/{id}`
- **Method:** `PUT`
- **Content-Type:** `multipart/form-data`
- **Description:** Updates an existing zone.
- **Parameters:**
  - **id** (Path, Integer): Zone ID.
  - **data** (Part, JSON String): JSON string representing `ZoneRequest`.
  - **image** (Part, File, Optional): New zone image file.
- **Response:** JSON with updated zone details.

### Delete Zone

- **URL:** `/zones/{id}`
- **Method:** `DELETE`
- **Description:** Deletes a zone.
- **Parameters:**
  - **id** (Path, Integer): Zone ID.
- **Response:** Success message.

### 3. Images (/images)

#### Get Image

- **URL:** /images/{filename}
- **Method:** GET
- **Description:** Serves the image file.
- **Parameters:**
  - **filename** (Path, String): Name of the file.
- **Response:** Image binary.

#### Upload Image

- **URL:** /images/upload
- **Method:** POST
- **Content-Type:** multipart/form-data
- **Description:** Generic image upload endpoint.
- **Parameters:**
  - **image** (Part, File): Image file.
  - **type** (Query, String): Image type (PROFILE or ZONE).
- **Response:** JSON with uploaded filename and details.

#### Delete Image

- **URL:** /images/{filename}
  - **Method:** DELETE
  - **Description:** Deletes an image file.
  - **Parameters:**
    - **filename** (Path, String): Filename to delete.
    - **type** (Query, String): Image type (PROFILE or ZONE).
  - **Response:** Success message.
- 

### 4. OAuth2 (/oauth2)

#### Get Login URLs

- **URL:** /oauth2/login-urls
- **Method:** GET
- **Description:** Returns the login URLs for configured OAuth providers (Kakao, Google, Naver).
- **Response:**

```
{
  "success": true,
  "data": {
    "loginUrls": {
      "google": "...",
      "kakao": "...",
      "naver": "..."
    }
  }
}
```

```
}  
}  
}
```

## Success/Failure Callbacks

- **GET /oauth2/success**: Redirected here on successful login.
- **GET /oauth2/failure**: Redirected here on failed login.