

# YOW Point of Interest API Documentation

---

## Introduction

This document provides documentation for the YOW Point of Interest API. It allows for the management of organizations, users within those organizations, and Points of Interest (POIs) including their location, reviews, and usage statistics.

The API follows RESTful conventions and uses JSON for request and response bodies. UUIDs are used for unique identifiers. Timestamps are generally in OffsetDateTime format.

Authentication and authorization mechanisms are not detailed in this document but should be considered for production deployment (e.g., via Spring Security using OAuth2/OIDC with Keycloak, as hinted in the project dependencies).

The base path for all version 1 APIs is typically `/api/v1`.

---

## Organization API

Base Path: `/api/v1/organizations`

APIs for managing organizations.

---

### Create a new organization

- **POST** `/api/v1/organizations`
- **Description:** Creates a new organization. (From `@Operation(summary = "Create a new organization")`)
- **Request Body:** `(application/json)`
  - Schema: `OrganizationDTO`
    - `organizationId`: UUID (Output only, generated by server)
    - `orgName`: String (Required)
    - `orgCode`: String (Optional, unique)
    - `orgType`: String (Optional)
    - `createdAt`: OffsetDateTime (Output only, generated by server)
    - `isActive`: Boolean (Optional, defaults to `true` if not provided on creation)
  - Example:

```
{
  "orgName": "Tech Innovators Inc.",
  "orgCode": "TECHINC001",
  "orgType": "Technology",
  "isActive": true
}
```

- **Responses:**

- **201 Created:** Organization created successfully.
  - Body: **OrganizationDTO**
  - Example:

```
{
  "organizationId": "a1b2c3d4-e5f6-7890-1234-567890abcdef",
  "orgName": "Tech Innovators Inc.",
  "orgCode": "TECHINC001",
  "orgType": "Technology",
  "createdAt": "2024-06-18T10:00:00Z",
  "isActive": true
}
```

- **400 Bad Request:** Invalid input.

- **Example cURL:**

```
curl -X POST "http://localhost:8080/api/v1/organizations" -H
"Content-Type: application/json" -d '{
  "orgName": "Tech Innovators Inc.",
  "orgCode": "TECHINC001",
  "orgType": "Technology",
  "isActive": true
}'
```

---

## Get an organization by its ID

- **GET /api/v1/organizations/{id}**
- **Description:** Retrieves an organization by its unique ID. (From @Operation(summary = "Get an organization by its ID"))
- **Path Parameters:**
  - **id** (UUID, required): ID of the organization to be retrieved.
- **Responses:**
  - **200 OK:** Found the organization.
    - Body: **OrganizationDTO**
    - Example:

```
{
  "organizationId": "a1b2c3d4-e5f6-7890-1234-567890abcdef",
  "orgName": "Tech Innovators Inc.",
  "orgCode": "TECHINC001",
  "orgType": "Technology",
  "createdAt": "2024-06-18T10:00:00Z",
  "isActive": true
}
```

- **404 Not Found:** Organization not found.
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/organizations/a1b2c3d4-e5f6-7890-1234-567890abcdef"
```

---

## Get all organizations

- **GET /api/v1/organizations**
- **Description:** Retrieves a list of all organizations. (From `@Operation(summary = "Get all organizations")`)
- **Responses:**
  - **200 OK:** Successfully retrieved list of organizations.
    - Body: `List<OrganizationDTO>`
    - Example:

```
[
  {
    "organizationId": "a1b2c3d4-e5f6-7890-1234-567890abcdef",
    "orgName": "Tech Innovators Inc.",
    "orgCode": "TECHINC001",
    "orgType": "Technology",
    "createdAt": "2024-06-18T10:00:00Z",
    "isActive": true
  },
  {
    "organizationId": "b2c3d4e5-f6a7-8901-2345-678901bcdef0",
    "orgName": "Green Solutions Ltd.",
    "orgCode": "GREENSL002",
    "orgType": "Environmental",
    "createdAt": "2024-06-17T11:00:00Z",
    "isActive": false
  }
]
```

- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/organizations"
```

---

## Update an existing organization

- **PUT** `/api/v1/organizations/{id}`
- **Description:** Updates an existing organization. (From `@Operation(summary = "Update an existing organization")`)
- **Path Parameters:**
  - `id` (UUID, required): ID of the organization to be updated.
- **Request Body:** `(application/json)`
  - Schema: `OrganizationDTO` (Fields to be updated)
  - Example:

```
{
  "orgName": "Tech Innovators Group",
  "orgType": "Global Technology",
  "isActive": true
}
```

- **Responses:**
  - **200 OK:** Organization updated successfully.
    - Body: `OrganizationDTO` (updated representation)
  - **400 Bad Request:** Invalid input.
  - **404 Not Found:** Organization not found.
- **Example cURL:**

```
curl -X PUT "http://localhost:8080/api/v1/organizations/a1b2c3d4-e5f6-7890-1234-567890abcdef" -H "Content-Type: application/json" -d '{
  "orgName": "Tech Innovators Group",
  "orgType": "Global Technology",
  "isActive": true
}'
```

---

## Delete an organization by its ID

- **DELETE** `/api/v1/organizations/{id}`
- **Description:** Deletes an organization by its ID. (From `@Operation(summary = "Delete an organization by its ID")`)
- **Path Parameters:**
  - `id` (UUID, required): ID of the organization to be deleted.
- **Responses:**
  - **204 No Content:** Organization deleted successfully.
  - **404 Not Found:** Organization not found.
- **Example cURL:**

```
curl -X DELETE "http://localhost:8080/api/v1/organizations/a1b2c3d4-e5f6-7890-1234-567890abcdef"
```

---

## User API

Base Path: `/api/v1/users`

APIs for managing application users.

---

### Create a new user

- **POST** `/api/v1/users`
- **Description:** Creates a new user. (From `@Operation(summary = "Create a new user")`)
- **Request Body:** (`application/json`)
  - Schema: `AppUserDTO`
    - `userId`: UUID (Output only, generated by server)
    - `orgId`: UUID (Required, ID of the organization this user belongs to)
    - `username`: String (Required)
    - `email`: String (Optional)
    - `phone`: String (Optional)
    - `role`: String (Required, e.g., 'USER', 'ADMIN')
    - `isActive`: Boolean (Optional, defaults to `true` if not provided)
    - `createdAt`: OffsetDateTime (Output only, generated by server)
  - Example:

```
{
  "orgId": "a1b2c3d4-e5f6-7890-1234-567890abcdef",
  "username": "johndoe",
  "email": "johndoe@example.com",
  "phone": "123-456-7890",
  "role": "USER",
  "isActive": true
}
```

- **Responses:**
  - **201 Created:** User created successfully.
    - Body: `AppUserDTO`
    - Example:

```
{
  "userId": "b2c3d4e5-f6a7-8901-2345-abcdef123456",
  "orgId": "a1b2c3d4-e5f6-7890-1234-567890abcdef",
  "username": "johndoe",
  "email": "johndoe@example.com",
  "phone": "123-456-7890",
  "role": "USER",
  "isActive": true,
```

```
"createdAt": "2024-06-18T10:05:00Z"
}
```

- **400 Bad Request:** Invalid input (e.g., missing `orgId` or `username`).
- **Example cURL:**

```
curl -X POST "http://localhost:8080/api/v1/users" -H "Content-Type: application/json" -d '{
  "orgId": "a1b2c3d4-e5f6-7890-1234-567890abcdef",
  "username": "johndoe",
  "email": "johndoe@example.com",
  "phone": "123-456-7890",
  "role": "USER",
  "isActive": true
}'
```

---

## Get a user by their ID

- **GET** `/api/v1/users/{id}`
- **Description:** Retrieves a user by their unique ID. (From `@Operation(summary = "Get a user by their ID")`)
- **Path Parameters:**
  - `id` (UUID, required): ID of the user to be retrieved.
- **Responses:**
  - **200 OK:** Found the user.
    - Body: `AppUserDTO`
  - **404 Not Found:** User not found.
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/users/b2c3d4e5-f6a7-8901-2345-abcdef123456"
```

---

## Get a user by their username

- **GET** `/api/v1/users/username/{username}`
- **Description:** Retrieves a user by their username. (From `@Operation(summary = "Get a user by their username")`)
- **Path Parameters:**
  - `username` (String, required): Username of the user to be retrieved.
- **Responses:**
  - **200 OK:** Found the user.
    - Body: `AppUserDTO`
  - **404 Not Found:** User not found.

- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/users/username/johndoe"
```

---

## Get a user by their email address

- **GET** `/api/v1/users/email/{email}`
- **Description:** Retrieves a user by their email address. (From `@Operation(summary = "Get a user by their email address")`)
- **Path Parameters:**
  - **email** (String, required): Email address of the user to be retrieved. (Ensure URL encoding if email contains special characters).
- **Responses:**
  - **200 OK:** Found the user.
    - Body: `AppUserDTO`
  - **404 Not Found:** User not found.
- **Example cURL:**

```
curl -X GET  
"http://localhost:8080/api/v1/users/email/johndoe%40example.com"
```

---

## Get all users

- **GET** `/api/v1/users`
- **Description:** Retrieves a list of all users. (From `@Operation(summary = "Get all users")`)
- **Responses:**
  - **200 OK:** Successfully retrieved list of users.
    - Body: `List<AppUserDTO>`
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/users"
```

---

## Update an existing user

- **PUT** `/api/v1/users/{id}`
- **Description:** Updates an existing user. (From `@Operation(summary = "Update an existing user")`)
- **Path Parameters:**
  - **id** (UUID, required): ID of the user to be updated.
- **Request Body:** (`application/json`)

- Schema: **AppUserDTO** (Fields to be updated. **orgId** can be included if changing organization. Password should be handled via a separate endpoint).
- Example:

```
{
  "email": "john.doe.updated@example.com",
  "phone": "987-654-3210",
  "isActive": false,
  "role": "ADMIN"
}
```

- **Responses:**

- **200 OK:** User updated successfully.
  - Body: **AppUserDTO**
- **400 Bad Request:** Invalid input.
- **404 Not Found:** User or specified **orgId** not found.

- **Example cURL:**

```
curl -X PUT "http://localhost:8080/api/v1/users/b2c3d4e5-f6a7-8901-2345-abcdef123456" -H "Content-Type: application/json" -d '{
  "email": "john.doe.updated@example.com",
  "phone": "987-654-3210",
  "isActive": false,
  "role": "ADMIN"
}'
```

---

## Delete a user by their ID

- **DELETE** `/api/v1/users/{id}`
- **Description:** Deletes a user by their ID. (From `@Operation(summary = "Delete a user by their ID")`)
- **Path Parameters:**
  - **id** (UUID, required): ID of the user to be deleted.
- **Responses:**
  - **204 No Content:** User deleted successfully.
  - **404 Not Found:** User not found.
- **Example cURL:**

```
curl -X DELETE "http://localhost:8080/api/v1/users/b2c3d4e5-f6a7-8901-2345-abcdef123456"
```

---

## Point of Interest API



Base Path: `/api/v1/pois`

APIs for managing Points of Interest (POIs).

---

## Create a new Point of Interest

- **POST** `/api/v1/pois`
- **Description:** Creates a new Point of Interest. (From `@Operation(summary = "Create a new Point of Interest")`)
- **Request Body:** (`application/json`)
  - Schema: `PointOfInterestDTO`
    - `poiId`: UUID (Output only)
    - `orgId`: UUID (Required, ID of the parent organization)
    - `townId`: UUID (Optional, ID of the town)
    - `createdByUserId`: UUID (Required, ID of the user creating the POI)
    - `poiName`: String (Required)
    - `poiType`: String (Required, e.g., 'RESTAURANT', 'HOTEL')
    - `poiCategory`: String (Required, e.g., 'Food & Drink', 'Landmark')
    - `poiLongName`: String (Optional)
    - `poiShortName`: String (Optional)
    - `poiFriendlyName`: String (Optional)
    - `poiDescription`: String (Optional)
    - `poiImages`: List (Optional, URLs to images)
    - `latitude`: Double (Required, geographical latitude)
    - `longitude`: Double (Required, geographical longitude)
    - `locationGeogWKT`: String (Output only, WKT representation of the location)
    - `poiAddress`: `AddressTypeDTO` (Optional, structured address)
    - `phoneNumber`: String (Optional)
    - `websiteUrl`: String (Optional)
    - `poiAmenities`: List (Optional, e.g., ["Wi-Fi", "Parking"])
    - `poiKeywords`: List (Optional, e.g., ["landmark", "historic"])
    - `poiTypeTags`: List (Optional, additional type tags)
    - `operationTimePlan`: String (Optional, JSON string representing opening hours, e.g., `{"Monday": "09:00-18:00"}`)
    - `poiContacts`: List<`ContactPersonTypeDTO`> (Optional)
    - `popularityScore`: Float (Output only, defaults to 0.0)
    - `isActive`: Boolean (Optional, defaults to `true` if not provided)
    - `deactivationReason`: String (Output only, set if POI is deactivated)
    - `deactivatedByUserId`: UUID (Output only)
    - `createdAt`: OffsetDateTime (Output only)
    - `updatedByUserId`: UUID (Optional for creation, usually for update)
    - `updatedAt`: OffsetDateTime (Output only)
  - Example:

```
{  
  "orgId": "a1b2c3d4-e5f6-7890-1234-567890abcdef",  
  /
```

```

"townId": "t1ownUuiD-town-uuid-town-uuid00001",
"createdByUserId": "b2c3d4e5-f6a7-8901-2345-abcdef123456",
"poiName": "Historic Grand Hotel",
"poiType": "HOTEL",
"poiCategory": "Accommodation",
"poiDescription": "A historic hotel offering luxury stays.",
"latitude": 48.8584,
"longitude": 2.2945,
"poiAddress": {
  "streetNumber": "123",
  "streetName": "Main Street",
  "city": "Paris",
  "stateProvince": "Île-de-France",
  "postalCode": "75001",
  "country": "France"
},
"phoneNumber": "+33 1 23 45 67 89",
"websiteUrl": "http://www.grandhotelhistoric.com",
"poiAmenities": ["Free WiFi", "Restaurant", "Valet Parking"],
"isActive": true
}

```

- **Responses:**

- **201 Created:** POI created successfully.
  - Body: **PointOfInterestDTO**
  - Example:

```

{
  "poiId": "p1o2i3d4-p5o6-7890-1234-poi012345poi",
  "orgId": "a1b2c3d4-e5f6-7890-1234-567890abcdef",
  "townId": "t1ownUuiD-town-uuid-town-uuid00001",
  "createdByUserId": "b2c3d4e5-f6a7-8901-2345-abcdef123456",
  "poiName": "Historic Grand Hotel",
  "poiType": "HOTEL",
  "poiCategory": "Accommodation",
  "poiDescription": "A historic hotel offering luxury
stays.",
  "latitude": 48.8584,
  "longitude": 2.2945,
  "locationGeogWKT": "POINT(2.2945 48.8584)",
  "poiAddress": {
    "streetNumber": "123",
    "streetName": "Main Street",
    "city": "Paris",
    "stateProvince": "Île-de-France",
    "postalCode": "75001",
    "country": "France"
  },
  "phoneNumber": "+33 1 23 45 67 89",
  "websiteUrl": "http://www.grandhotelhistoric.com",
  "poiAmenities": ["Free WiFi", "Restaurant", "Valet

```

```
Parking"],
  "isActive": true,
  "createdAt": "2024-06-18T10:10:00Z",
  "updatedAt": "2024-06-18T10:10:00Z",
  "popularityScore": 0.0
}
```

- **400 Bad Request**: Invalid input.

- **Example cURL:**

```
curl -X POST "http://localhost:8080/api/v1/pois" -H "Content-Type: application/json" -d '{
  "orgId": "a1b2c3d4-e5f6-7890-1234-567890abcdef",
  "createdByUserId": "b2c3d4e5-f6a7-8901-2345-abcdef123456",
  "poiName": "Historic Grand Hotel",
  "poiType": "HOTEL",
  "poiCategory": "Accommodation",
  "latitude": 48.8584,
  "longitude": 2.2945,
  "poiAddress": {"city": "Paris", "country": "France"}
}'
```

---

## Get a Point of Interest by its ID

- **GET /api/v1/pois/{id}**
- **Description:** Retrieves a POI by its unique ID.
- **Path Parameters:**
  - **id** (UUID, required): ID of the POI.
- **Responses:**
  - **200 OK**: Found the POI.
    - Body: **PointOfInterestDTO**
  - **404 Not Found**: POI not found.
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/pois/p102i3d4-p506-7890-1234-poi012345poi"
```

---

## Get all Points of Interest (with filtering)

- **GET /api/v1/pois**
- **Description:** Retrieves a list of all POIs, with optional filtering.
- **Query Parameters:**
  - **name** (String, optional): Filter by POI name (case-insensitive, partial match).
  - **type** (String, optional): Filter by POI type.

- **category** (String, optional): Filter by POI category.
- **Responses:**
  - **200 OK:** Successfully retrieved list of POIs.
    - Body: `List<PointOfInterestDTO>`
- **Example cURL (filter by type):**

```
curl -X GET "http://localhost:8080/api/v1/pois?type=HOTEL"
```

---

## Update an existing Point of Interest

- **PUT /api/v1/pois/{id}**
- **Description:** Updates an existing POI.
- **Path Parameters:**
  - **id** (UUID, required): ID of the POI to be updated.
- **Request Body:** (`application/json`)
  - Schema: `PointOfInterestDTO`
  - Example:

```
{
  "poiName": "Historic Grand Hotel - Renovated",
  "poiDescription": "A historic hotel, now fully renovated,
offering luxury stays and fine dining.",
  "latitude": 48.8585,
  "longitude": 2.2946,
  "phoneNumber": "+33 1 23 45 67 99",
  "updatedByUserId": "b2c3d4e5-f6a7-8901-2345-abcdef123456",
  "isActive": true
}
```

- **Responses:**
  - **200 OK:** POI updated successfully.
    - Body: `PointOfInterestDTO`
  - **400 Bad Request:** Invalid input.
  - **404 Not Found:** POI or related user ID not found.
- **Example cURL:**

```
curl -X PUT "http://localhost:8080/api/v1/pois/p102i3d4-p5o6-7890-1234-poi012345poi" -H "Content-Type: application/json" -d '{
  "poiName": "Historic Grand Hotel - Renovated",
  "latitude": 48.8585,
  "longitude": 2.2946,
  "updatedByUserId": "b2c3d4e5-f6a7-8901-2345-abcdef123456"
}'
```

## Delete a Point of Interest by its ID

- **DELETE** `/api/v1/pois/{id}`
- **Description:** Deletes a POI by its ID.
- **Path Parameters:**
  - `id` (UUID, required): ID of the POI to be deleted.
- **Responses:**
  - **204 No Content:** POI deleted successfully.
  - **404 Not Found:** POI not found.
- **Example cURL:**

```
curl -X DELETE "http://localhost:8080/api/v1/pois/p102i3d4-p5o6-7890-1234-poi012345poi"
```

---

## Find Nearby Points of Interest

- **GET** `/api/v1/pois/nearby`
- **Description:** Finds POIs within a certain distance (in meters) from a given geographic point.
- **Query Parameters:**
  - `longitude` (Double, required): Longitude of the center point.
  - `latitude` (Double, required): Latitude of the center point.
  - `distance` (Double, required): Search radius in meters.
- **Responses:**
  - **200 OK:** Successfully retrieved list of nearby POIs.
    - Body: `List<PointOfInterestDTO>`
  - **400 Bad Request:** Invalid input.
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/pois/nearby?longitude=2.2945&latitude=48.8584&distance=1000"
```

---

## Get Points of Interest by Town ID

- **GET** `/api/v1/pois/town/{townId}`
- **Description:** Retrieves POIs for a specific Town ID.
- **Path Parameters:**
  - `townId` (UUID, required): UUID of the town.
- **Responses:**
  - **200 OK:** POIs found.
    - Body: `List<PointOfInterestDTO>`
  - **404 Not Found:** No POIs found for the town ID.
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/pois/town/t1ownUuid-town-uuid-town-uuid00001"
```

---

## Get Points of Interest by State/Province

- **GET** `/api/v1/pois/state/{stateProvince}`
- **Description:** Retrieves POIs by state or province.
- **Path Parameters:**
  - `stateProvince` (String, required): Name of the state/province.
- **Responses:**
  - **200 OK:** POIs found (can be an empty list).
    - Body: `List<PointOfInterestDTO>`
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/pois/state/Ile-de-France"
```

---

## Get Points of Interest by Creator User ID

- **GET** `/api/v1/pois/user/{userId}`
- **Description:** Retrieves POIs created by a specific user.
- **Path Parameters:**
  - `userId` (UUID, required): UUID of the user.
- **Responses:**
  - **200 OK:** POIs found.
    - Body: `List<PointOfInterestDTO>`
  - **404 Not Found:** No POIs found for the user.
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/pois/user/b2c3d4e5-f6a7-8901-2345-abcdef123456"
```

---

## Get all Points of Interest ordered by popularity

- **GET** `/api/v1/pois/popular`
- **Description:** Retrieves all POIs, ordered by popularity score (descending).
- **Responses:**
  - **200 OK:** Successfully retrieved POIs.
    - Body: `List<PointOfInterestDTO>`
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/pois/popular"
```

---

Get active Points of Interest ordered by popularity

- **GET** `/api/v1/pois/active/popular`
- **Description:** Retrieves active POIs, ordered by popularity score (descending).
- **Responses:**
  - **200 OK:** Successfully retrieved POIs.
    - Body: `List<PointOfInterestDTO>`
- **Example cURL:**

```
curl -X GET "http://localhost:8080/api/v1/pois/active/popular"
```

---