

Ain El Watan API Documentation

Overview

The Ain El Watan API provides endpoints for user authentication via OTP verification and report management. This document outlines the available endpoints, request parameters, and expected responses.

Base URL

`http://127.0.0.1:8000/api`

Authentication

Most endpoints require authentication using a Bearer token, which is obtained after successful OTP verification.

Header Format

`Authorization: Bearer {token}`

Endpoints

1. Send OTP

Sends a one-time password to the specified phone number for authentication.

- **URL:** `/send-otp`
- **Method:** `POST`
- **Authentication Required:** No

Request Body

Parameter	Type	Required	Description
national_id	String	Yes	User's national ID (14 digits)
phone	String	Yes	User's phone number (e.g., 01000000000)

Example Request

json

```
{
  "national_id": "12345678901234",
  "phone": "01157861819"
}
```

Example Response

json

```
{
  "status": "success",
  "message": "OTP sent successfully"
}
```

2. Verify OTP

Verifies the OTP entered by the user and returns an authentication token upon success.

- **URL:** `/verify-otp`
- **Method:** `POST`
- **Authentication Required:** No

Request Body

Parameter	Type	Required	Description
phone	String	Yes	User's phone number
otp	String	Yes	The OTP received by the user

Example Request

json

```
{
  "phone": "01157861819",
  "otp": "2441"
}
```

Example Response

json

```
{
  "status": "success",
  "token": "4|suwGAm1Ma0VuDHFcNIFAWQTCEdZTCJzX1UXPaSXc",
  "user": {
    "id": 1,
    "name": "User Name",
    "phone": "01000000000"
  }
}
```

3. Get Report Types

Retrieves a list of available report types.

- **URL:** `/report-types`
- **Method:** `GET`
- **Authentication Required:** No

Example Response

json

```
{
  "status": "success",
  "data": [
    {
      "id": 1,
      "name": "Maintenance Request",
      "created_at": "2023-09-15T10:30:00.000000Z"
    },
    {
      "id": 2,
      "name": "Complaint",
      "created_at": "2023-09-15T10:30:00.000000Z"
    }
  ]
}
```

4. Create User Report

Submits a new report with details and optional file attachment.

- **URL:** `/reports`
- **Method:** `POST`
- **Authentication Required:** Yes (Bearer Token)

Request Parameters

Parameter	Type	Required	Description
description	String	Yes	Detailed description of the report
ll	String	Yes	Latitude coordinate
lg	String	Yes	Longitude coordinate
location	String	Yes	Text description of the location
report_type_id	Integer	Yes	ID of the report type (from report-types endpoint)
file	File	No	File attachment (PDF, image, etc.)

Example Request

```
// Form data
description: "هناك حفرة كبيرة جدا في الطريق"
ll: "20.589"
lg: "20.589"
location: "ش المعز لدين الله 16"
report_type_id: "1"
file: [binary file data]
```

Example Response

json

```
{
  "status": "success",
  "message": "Report submitted successfully",
  "data": {
    "id": 123,
    "description": "هناك حفرة كبيرة جدا في الطريق",
    "location": "ش المعز لدين الله 16",
    "coordinates": {
      "latitude": "20.589",
      "longitude": "20.589"
    },
    "created_at": "2023-09-15T11:30:00.000000Z",
    "report_type": {
      "id": 1,
      "name": "Maintenance Request"
    },
    "file_url": "http://127.0.0.1:8000/storage/reports/file_123.pdf"
  }
}
```

5. Get User Reports

Retrieves all reports submitted by the authenticated user.

- **URL:** `/reports`
- **Method:** `GET`
- **Authentication Required:** Yes (Bearer Token)

Example Response

json

```
{
  "status": "success",
  "data": [
    {
      "id": 123,
      "description": "و قد تفضل جلالتہ و وافق علي المطالبين",
      "location": "ش المعز لدين الله 16",
      "coordinates": {
        "latitude": "20.589",
        "longitude": "20.589"
      },
      "created_at": "2023-09-15T11:30:00.000000Z",
      "report_type": {
        "id": 1,
        "name": "Maintenance Request"
      },
      "file_url": "http://127.0.0.1:8000/storage/reports/file_123.pdf"
    },
    {
      "id": 124,
      "description": "Report description here",
      "location": "Another location",
      "coordinates": {
        "latitude": "21.589",
        "longitude": "22.589"
      },
      "created_at": "2023-09-16T10:30:00.000000Z",
      "report_type": {
        "id": 2,
        "name": "Complaint"
      },
      "file_url": null
    }
  ]
}
```

Error Handling

The API returns appropriate HTTP status codes along with error messages for failed requests:

- `400 Bad Request`: Invalid input parameters

- `401 Unauthorized`: Authentication failed or token expired
- `404 Not Found`: Requested resource not found
- `500 Internal Server Error`: Server-side error

Example Error Response

json

```
{
  "status": "error",
  "message": "Invalid OTP provided",
  "errors": {
    "otp": ["The OTP entered is incorrect or has expired"]
  }
}
```

Notes

- All dates are returned in ISO 8601 format (YYYY-MM-DDThh:mm:ss.sssZ)
- File uploads support PDF, JPG, PNG, and other common formats
- The token received after OTP verification should be included in all authenticated requests