National Eye API Documentation

Overview

The National Eye API provides a comprehensive system for managing reports, procedures, and report types. It includes user authentication via OTP verification, full CRUD operations for report types and procedures, and advanced report management capabilities.

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}
```

API Endpoints

Authentication

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

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": "5511"
}
```

Example Response:

json

```
{
  "status": "success",
  "token": "4|suwGAm1Ma0VuDHFcNIFAWQTCEdZTCJzX1UXPaSXc",
  "user": {
    "id": 1,
    "name": "User Name",
    "phone": "01157861819"
  }
}
```

Report Types Management

3. Get All Report Types

Retrieves a list of all available report types.

- **URL**: `/report-types`
- **Method**: `GET`
- **Authentication Required**: Yes

Example Response:

json

```
{
  "status": "success",
  "data": [
    {
      "id": 1,
      "name": "مشاكل في الطريق",
      "notes": "",
      "created_at": "2023-09-15T10:30:00.000000Z"
    },
    {
      "id": 2,
      "name": "حفرة في الطريق",
      "notes": "الابلاغ عن حفرة في الطريق",
      "created_at": "2023-09-15T10:30:00.000000Z"
    }
  ]
}
```

4. Create Report Type

Creates a new report type.

- **URL**: `/report-types`
- **Method**: `POST`
- **Authentication Required**: Yes

Request Body:

Parameter	Type	Required	Description
name	String	Yes	Name of the report type
notes	String	No	Additional notes about the report type

Example Request:

```json
{
  "name": "مشاكل في الطريق",
  "notes": ""
}
```

5. Show Report Type

Retrieves details of a specific report type.

- **URL**: `/report-types/{id}`
- **Method**: `GET`
- **Authentication Required**: Yes

Example Response:

json

```
{
  "status": "success",
  "data": {
    "id": 3,
    "name": "مشاكل في الطريق",
    "notes": "",
    "created_at": "2023-09-15T10:30:00.000000Z"
  }
}
```

6. Update Report Type

Updates an existing report type.

- **URL**: `/report-types/{id}`
- **Method**: `PUT`
- **Authentication Required**: Yes

Request Body:

Parameter	Type	Required	Description
name	String	Yes	Updated name of the report type
notes	String	No	Updated notes

Example Request:

json

```
{
  "name": "حفرة في الطريق",
  "notes": "الابلاغ عن حفرة في الطريق"
}
```

7. Delete Report Type

Deletes a specific report type.

- **URL**: `/report-types/{id}`
- **Method**: `DELETE`

- **Authentication Required**: Yes

Procedures Management

8. Get All Procedures

Retrieves a list of all available procedures.

- **URL**: `/procedures`
- **Method**: `GET`
- **Authentication Required**: Yes

Example Response:

```json
{
  "status": "success",
  "data": [
    {
      "id": 1,
      "name": "Test",
      "description": "Test",
      "created_at": "2023-09-15T10:30:00.000000Z"
    }
  ]
}
```

9. Create Procedure

Creates a new procedure.

- **URL**: `/procedures`
- **Method**: `POST`
- **Authentication Required**: Yes

Request Body:

Parameter	Type	Required	Description
name	String	Yes	Name of the procedure
description	String	Yes	Description of the procedure

Example Request:

json

```
{
  "name": "Test",
  "description": "Test"
}
```

10. Show Procedure

Retrieves details of a specific procedure.

- **URL**: `/procedures/{id}`
- **Method**: `GET`
- **Authentication Required**: Yes

11. Update Procedure

Updates an existing procedure.

- **URL**: `/procedures/{id}`
- **Method**: `PUT`
- **Authentication Required**: Yes

Request Body:

Parameter	Type	Required	Description
name	String	Yes	Updated name of the procedure
description	String	Yes	Updated description

Example Request:

json

```
{
  "name": "Test2",
  "description": "Test2"
}
```

12. Delete Procedure

Deletes a specific procedure.

- **URL**: `/procedures/{id}`
- **Method**: `DELETE`
- **Authentication Required**: Yes

Report Procedures Management

13. Get Report Procedures

Retrieves all procedures associated with a specific report.

- **URL**: `/reports/{report_id}/procedures`
- **Method**: `GET`
- **Authentication Required**: Yes

14. Create Report Procedure

Associates a procedure with a specific report.

- **URL**: `/report/{report_id}/procedures`
- **Method**: `POST`
- **Authentication Required**: Yes

Request Body:

Parameter	Type	Required	Description
procedure_id	Integer	Yes	ID of the procedure to associate
notes	String	No	Additional notes about the procedure

Example Request:

json

```
{
  "procedure_id": 3,
  "notes": "تم اتخاذ الإجراء اللازم"
}
```

15. Update Report Procedure

Updates the notes for a procedure associated with a report.

- **URL**: `/report/{report_id}/procedures/{procedure_id}`
- **Method**: `PUT`
- **Authentication Required**: Yes

Request Body:

Parameter	Type	Required	Description
notes	String	Yes	Updated notes

Example Request:

```json
{
  "notes": "تم التدخل و الحل حالا"
}
```

16. Delete Report Procedure

Removes a procedure association from a report.

- **URL**: `/report/{report_id}/procedures/{procedure_id}`
- **Method**: `DELETE`
- **Authentication Required**: Yes

Reports Management

17. Get All Reports

Retrieves all reports in the system (admin function).

- **URL**: `/all-reports`
- **Method**: `GET`
- **Authentication Required**: Yes

18. Create User Report

Submits a new report with details and optional file attachment.

- **URL**: `/reports`
- **Method**: `POST`

- **Authentication Required**: Yes

Request Parameters (Form Data):

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
report_no	String	Yes	Report reference number
file	File	No	File attachment (PDF, image, XLSX, MP4, MP3, etc.)

Example Request:

```
description: "هناك حفرة كبيرة جدا في الطريق"
ll: "20.589"
lg: "20.589"
location: "16 ش المعز لدين الله"
report_type_id: "2"
report_no: "REP-202567"
file: [binary file data]
```

19. Get User Reports

Retrieves all reports submitted by the authenticated user.

- **URL**: `/reports`
- **Method**: `GET`
- **Authentication Required**: Yes

Example Response:

```json
{
  "status": "success",
  "data": [
    {
      "id": 123,
      "report_no": "REP-202567",
      "description": "هناك حفرة كبيرة جدا في الطريق",
      "location": "16 ش المعز لدين الله",
      "coordinates": {
        "latitude": "20.589",
        "longitude": "20.589"
      },
      "created_at": "2023-09-15T11:30:00.000000Z",
      "report_type": {
        "id": 2,
        "name": "حفرة في الطريق"
      },
      "file_url": "http://127.0.0.1:8000/storage/reports/file_123.pdf"
    }
  ]
}
```

20. Get Report Details

Retrieves detailed information about a specific report.

- **URL**: `/reports/{id}`
- **Method**: `GET`
- **Authentication Required**: Yes

Example Response:

```json
{
  "status": "success",
  "data": {
    "id": 3,
    "report_no": "REP-202567",
    "description": "هناك حفرة كبيرة جدا في الطريق",
    "location": "16 ش المعز لدين الله",
    "coordinates": {
      "latitude": "20.589",
      "longitude": "20.589"
    },
    "created_at": "2023-09-15T11:30:00.000000Z",
    "report_type": {
      "id": 2,
      "name": "حفرة في الطريق",
      "notes": "الابلاغ عن حفرة في الطريق"
    },
    "procedures": [
      {
        "id": 1,
        "name": "Test",
        "description": "Test",
        "notes": "تم اتخاذ الإجراء اللازم"
      }
    ],
    "file_url": "http://127.0.0.1:8000/storage/reports/file_123.pdf"
  }
}
```

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
- `403 Forbidden`: Insufficient permissions
- `404 Not Found`: Requested resource not found
- `422 Unprocessable Entity`: Validation errors
- `500 Internal Server Error`: Server-side error

Example Error Response

json

```json
{
  "status": "error",
  "message": "Validation failed",
  "errors": {
    "name": ["The name field is required"],
    "description": ["The description field must be at least 3 characters"]
  }
}
```

Authentication Flow

1. **Send OTP**: Submit national ID and phone number to receive OTP
2. **Verify OTP**: Submit phone number and OTP to receive authentication token
3. **Use Token**: Include the Bearer token in all subsequent API requests

Data Models

Report Type

- `id`: Unique identifier
- `name`: Name of the report type (Arabic text supported)
- `notes`: Optional additional information
- `created_at`: Creation timestamp

Procedure

- `id`: Unique identifier
- `name`: Name of the procedure
- `description`: Detailed description of the procedure
- `created_at`: Creation timestamp

Report

- `id`: Unique identifier
- `report_no`: Unique report reference number
- `description`: Detailed description (Arabic text supported)

- `location`: Text description of location
- `coordinates`: Latitude and longitude
- `report_type_id`: Associated report type
- `file_url`: URL to uploaded file (if any)
- `created_at`: Creation timestamp

Notes

- All dates are returned in ISO 8601 format
- File uploads support PDF, JPG, PNG, XLSX, MP4, MP3, and other common formats
- Arabic text is fully supported throughout the API
- Report numbers should follow the format "REP-XXXXXX"
- Coordinates should be provided as decimal degrees