

Eyecon AI API Documentation

Version 1.0.0

May 15, 2025



AI-Powered Video Surveillance and Anomaly Detection System

Contents

1	Introduction	2
2	Authentication	2
3	Base URL	2
4	API Endpoints	2
4.1	Auth	2
4.1.1	Login	2
4.2	Admin	3
4.2.1	Create Organization Admin	3
4.2.2	Create New Organization	3
4.2.3	Update Organization	4
4.2.4	Get All Organizations	4
4.2.5	Get Organization By ID	4
4.2.6	Set Max Camera Limit	5
4.3	Analytics	5
4.3.1	Anomaly Analytics	5
4.4	Organization Admin	6
4.4.1	Camera Management	6
4.4.2	Anomaly Management	10
4.4.3	Organization Management	11
5	Request and Response Formats	12
6	Error Codes	12
7	Changelog	12
7.1	Version 1.0.0 (May 15, 2025)	12

1 Introduction

The Eyecon AI API provides a comprehensive set of endpoints for AI-powered video surveillance and anomaly detection systems. This RESTful API allows organizations to manage cameras, detect anomalies, and analyze surveillance data using advanced AI models.

2 Authentication

All API requests (except login) require authentication using a Bearer token in the Authorization header.

Example:

```
Authorization: Bearer <ACCESS_TOKEN>
```

To obtain an access token, use the login endpoint.

3 Base URL

Replace «SERVER_URL» with your actual server URL.

```
<<SERVER_URL>>/api
```

4 API Endpoints

4.1 Auth

4.1.1 Login

Authenticates a user and returns an access token.

- **URL:** /auth/login
- **Method:** POST
- **Auth Required:** No
- **Content-Type:** application/json

Request Body:

```
{
  "username": "username",
  "password": "password"
}
```

Success Response:

- **Code:** 200 OK

Content Example:

```
{
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "id": 1,
    "username": "username",
    "name": "User Name",
    "role": "ORGANIZATION_ADMIN"
  }
}
```

4.2 Admin

4.2.1 Create Organization Admin

Creates a new admin for a specific organization.

- **URL:** /admin/organization/{organizationId}/admin
- **Method:** POST
- **Auth Required:** Yes
- **Content-Type:** application/json

URL Parameters:

- **organizationId:** ID of the organization

Request Body:

```
{
  "username": "admin",
  "name": "Admin Name",
  "email": "admin@example.com",
  "password": "password"
}
```

Success Response:

- **Code:** 201 Created

4.2.2 Create New Organization

Creates a new organization.

- **URL:** /admin/organization
- **Method:** POST
- **Auth Required:** Yes
- **Content-Type:** application/json

Request Body:

```
{
  "name": "Organization Name",
  "maxCameras": 4
}
```

Success Response:

- **Code:** 201 Created

4.2.3 Update Organization

Updates an existing organization.

- **URL:** /admin/organization/{organizationId}
- **Method:** PUT
- **Auth Required:** Yes
- **Content-Type:** application/json

URL Parameters:

- **organizationId:** ID of the organization

Request Body:

```
{
  "name": "Updated Organization Name",
  "maxCameras": 10
}
```

Success Response:

- **Code:** 200 OK

4.2.4 Get All Organizations

Retrieves a list of all organizations.

- **URL:** /admin/organizations
- **Method:** GET
- **Auth Required:** Yes

Success Response:

- **Code:** 200 OK

4.2.5 Get Organization By ID

Retrieves details of a specific organization.

- **URL:** /admin/organization/{organizationId}
- **Method:** GET
- **Auth Required:** Yes

URL Parameters:

- **organizationId:** ID of the organization

Success Response:

- **Code:** 200 OK

4.2.6 Set Max Camera Limit

Updates the maximum number of cameras allowed for an organization.

- **URL:** /admin/organization/{organizationId}/max-cameras
- **Method:** PATCH
- **Auth Required:** Yes
- **Content-Type:** application/json

URL Parameters:

- **organizationId:** ID of the organization

Request Body:

```
{
  "maxCameras": 3
}
```

Success Response:

- **Code:** 200 OK

4.3 Analytics

4.3.1 Anomaly Analytics

Log Anomaly Records a new anomaly event.

- **URL:** /analytics/anomaly
- **Method:** POST
- **Auth Required:** Yes
- **Content-Type:** application/json

Success Response:

- **Code:** 201 Created

Get Anomalies Retrieves all anomalies.

- **URL:** /analytics/anomaly
- **Method:** GET
- **Auth Required:** Yes

Success Response:

- **Code:** 200 OK

Get Anomalies Stats Retrieves statistical information about detected anomalies.

- **URL:** /analytics/anomaly/stats
- **Method:** GET
- **Auth Required:** Yes

Success Response:

- **Code:** 200 OK

Get Daily Anomaly Counts Retrieves daily counts of anomalies.

- **URL:** /analytics/anomaly/daily-counts
- **Method:** GET
- **Auth Required:** Yes

Success Response:

- **Code:** 200 OK

Get Today's Anomaly Count Retrieves the count of anomalies detected today.

- **URL:** /analytics/anomaly/counts/today
- **Method:** GET
- **Auth Required:** Yes

Success Response:

- **Code:** 200 OK

Get All Anomalies for All Organizations Retrieves anomalies across all organizations.

- **URL:** /public/anomalies
- **Method:** GET
- **Auth Required:** Yes

Success Response:

- **Code:** 200 OK

4.4 Organization Admin

4.4.1 Camera Management

Get All Cameras Retrieves all cameras for a specific organization.

- **URL:** /organization/{organizationId}/cameras
- **Method:** GET
- **Auth Required:** Yes

URL Parameters:

- **organizationId:** ID of the organization

Success Response:

- **Code:** 200 OK

Get Online Cameras Retrieves all online cameras for a specific organization.

- **URL:** /organization/{organizationId}/cameras/online
- **Method:** GET
- **Auth Required:** Yes

URL Parameters:

- **organizationId:** ID of the organization

Success Response:

- **Code:** 200 OK

Add Camera Adds a new camera to an organization.

- **URL:** /organization/{organizationId}/camera
- **Method:** POST
- **Auth Required:** Yes
- **Content-Type:** application/json

URL Parameters:

- **organizationId:** ID of the organization

Request Body:

```
{
  "location": "Camera Location",
  "ipAddress": "http://camera-ip-address:port/video",
  "cameraType": "Camera Type",
  "cameraDescription": "Detailed description of camera location and surroundings"
}
```

Success Response:

- **Code:** 201 Created

Edit Camera Updates an existing camera.

- **URL:** /organization/{organizationId}/camera/{cameraId}
- **Method:** PUT
- **Auth Required:** Yes
- **Content-Type:** application/json

URL Parameters:

- **organizationId:** ID of the organization
- **cameraId:** ID of the camera

Request Body:

```
{
  "location": "Updated Location",
  "ipAddress": "Updated IP",
  "cameraType": "Updated Type",
  "cameraDescription": "Updated Description",
  "normalConditions": [
    {
      "conditionId": 64,
      "description": "a photo of person walking through door"
    },
    {
      "conditionId": 65,
      "description": "a photo of student showing id card"
    }
  ]
}
```

Success Response:

- Code: 200 OK

Delete Camera Removes a camera from an organization.

- URL: /organization/{organizationId}/camera/{cameraId}
- Method: DELETE
- Auth Required: Yes

URL Parameters:

- organizationId: ID of the organization
- cameraId: ID of the camera

Success Response:

- Code: 204 No Content

Get Anomaly Stats Retrieves anomaly statistics for cameras in an organization.

- URL: /organization/{organizationId}/camera/anomaly-stats
- Method: GET
- Auth Required: Yes

URL Parameters:

- organizationId: ID of the organization

Success Response:

- Code: 200 OK

Get Comprehensive Camera Details Retrieves detailed information about all cameras in an organization.

- **URL:** /organization/{organizationId}/cameras/comprehensive
- **Method:** GET
- **Auth Required:** Yes

URL Parameters:

- **organizationId:** ID of the organization

Success Response:

- **Code:** 200 OK

Get Normal Conditions for All Cameras Retrieves normal conditions configured for all cameras in an organization.

- **URL:** /organization/{organizationId}/normal-conditions
- **Method:** GET
- **Auth Required:** Yes

URL Parameters:

- **organizationId:** ID of the organization

Success Response:

- **Code:** 200 OK

Get Cameras with Anomalies Retrieves cameras that have detected anomalies.

- **URL:** /organization/{organizationId}/cameras/anomalies
- **Method:** GET
- **Auth Required:** Yes

URL Parameters:

- **organizationId:** ID of the organization

Success Response:

- **Code:** 200 OK

Get Comprehensive Camera Details (Public) Retrieves detailed camera information through a public endpoint.

- **URL:** /public/organization/cameras/comprehensive
- **Method:** GET
- **Auth Required:** No

Success Response:

- **Code:** 200 OK

4.4.2 Anomaly Management

Add Anomaly Adds a new anomaly detection rule.

- **URL:** /organization/{organizationId}/anomaly
- **Method:** POST
- **Auth Required:** Yes
- **Content-Type:** application/json

URL Parameters:

- **organizationId:** ID of the organization

Request Body:

```
{
  "title": "Anomaly Title",
  "description": "Detailed description of the anomaly",
  "criticality": "Critical",
  "modelName": "VLM",
  "cameraIds": [29],
  "startTime": "22:00:00",
  "endTime": "06:00:00",
  "daysOfWeek": ["MON", "TUE", "WED", "THU", "FRI"]
}
```

Success Response:

- **Code:** 201 Created

Get All Anomalies Retrieves all anomaly rules for an organization.

- **URL:** /organization/{organizationId}/anomalies
- **Method:** GET
- **Auth Required:** Yes

URL Parameters:

- **organizationId:** ID of the organization

Success Response:

- **Code:** 200 OK

Edit Anomaly Updates an existing anomaly rule.

- **URL:** /organization/{organizationId}/anomaly/{anomalyId}
- **Method:** PUT
- **Auth Required:** Yes
- **Content-Type:** application/json

URL Parameters:

- **organizationId:** ID of the organization
- **anomalyId:** ID of the anomaly

Request Body:

```
{
  "title": "Updated Title",
  "description": "Updated description",
  "criticality": "Critical",
  "modelName": "VLM",
  "cameraIds": [29],
  "startTime": "22:00:00",
  "endTime": "06:00:00",
  "daysOfWeek": ["MON", "TUE", "WED", "THU", "FRI"]
}
```

Success Response:

- **Code:** 200 OK

Delete Anomaly Removes an anomaly rule.

- **URL:** /organization/{organizationId}/anomaly/{anomalyId}
- **Method:** DELETE
- **Auth Required:** Yes

URL Parameters:

- **organizationId:** ID of the organization
- **anomalyId:** ID of the anomaly

Success Response:

- **Code:** 204 No Content

4.4.3 Organization Management

Update Organization Updates an organization's details.

- **URL:** /organization/{organizationId}
- **Method:** PUT
- **Auth Required:** Yes
- **Content-Type:** application/json

URL Parameters:

- **organizationId:** ID of the organization

Request Body:

```
{
  "name": "Updated Organization Name"
}
```

Success Response:

- **Code:** 200 OK

Get My Organization Details Retrieves details of the current user's organization.

- **URL:** /organization/{organizationId}/details
- **Method:** GET
- **Auth Required:** Yes

URL Parameters:

- **organizationId:** ID of the organization

Success Response:

- **Code:** 200 OK

5 Request and Response Formats

- All API requests and responses use JSON format.
- Timestamps are in ISO 8601 format (YYYY-MM-DDThh:mm:ss.sssZ).
- Time values are in 24-hour format (HH:MM:SS).

6 Error Codes

Code	Description
400	Bad Request - Invalid request format or parameters
401	Unauthorized - Authentication is required
403	Forbidden - Insufficient permissions
404	Not Found - The requested resource does not exist
409	Conflict - Resource already exists
422	Unprocessable Entity - Validation errors
500	Internal Server Error

Table 1: API Error Codes and Descriptions

7 Changelog

7.1 Version 1.0.0 (May 15, 2025)

- Initial release of the Eyecon AI API