Contents

[2. Authentication 1](#_Toc166537592)

[3. Error Handling 2](#_Toc166537593)

[4. Endpoints 2](#_Toc166537594)

[a. Get all Animals 2](#_Toc166537595)

[b. Get Pre-processed Animals 2](#_Toc166537596)

[c. Add Owner 2](#_Toc166537597)

[d. Get all Owners 3](#_Toc166537598)

[e. Get Owner by ID 3](#_Toc166537599)

[f. Update Owner by ID 3](#_Toc166537600)

[g. Delete Owner by ID 3](#_Toc166537601)

[h. Get all Animal Categories 4](#_Toc166537602)

[i. Add Animal to Animal Category 4](#_Toc166537603)

[j. Update Animal Category by ID 4](#_Toc166537604)

[k. Delete Animal Category by ID 5](#_Toc166537605)

[l. Add System 5](#_Toc166537606)

[m. Login System 5](#_Toc166537607)

[II. Authentication API Documentation 6](#_Toc166537608)

[1. Base URL 6](#_Toc166537609)

[2. Endpoints 6](#_Toc166537610)

[a. Register User 6](#_Toc166537611)

[b. Login User 6](#_Toc166537612)

1. **Animal Management API Documentation**

This API provides endpoints to manage animals, owners, animal categories, and systems.

1. **Base URL**

<http://your-base-url.com/api>

## Authentication

Some endpoints require authentication using JWT. Include the token in the **Authorization** header of your request.

## Error Handling

If an error occurs, the API will return an appropriate HTTP status code along with a JSON response containing an error message.

## Endpoints

### Get all Animals

Bash: GET /getanimal

Get all animals from the database.

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Array of animal objects

### Get Pre-processed Animals

Bash: GET /getanimals

Get pre-processed animals.

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Array of preprocessed animal objects

### Add Owner

Bash: POST /addOwner

Add a new owner to the database.

**Request Body**

* **cid** (string): Owner's CID
* **ownername** (string): Owner's name
* **contactNo** (string): Owner's contact number
* **emailID** (string): Owner's email ID

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message

### Get all Owners

Bash: GET /getOwner

Get all owners from the database.

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Array of owner objects

### Get Owner by ID

Bash: GET /getOwner/:id

Get owner by ID from the database.

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Owner object

### Update Owner by ID

Bash: POST /updateOwner/:id

Update owner by ID in the database.

**Request Body**

* Same as Add Owner

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message

### Delete Owner by ID

Bash: DELETE /deleteOwner/:id

Delete owner by ID from the database.

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message

### Get all Animal Categories

Bash: GET /getAnimalCat

Get all animal categories from the database.

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Array of animal category objects

### Add Animal to Animal Category

Bash: POST /addAnimal

Add an animal to an animal category in the database.

**Request Body**

* **acid** (string): Animal category ID
* **animalname** (string): Animal's name
* **animaldescription** (string): Animal's description

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message

### Update Animal Category by ID

Bash: POST /updateAnimalCat/:id

Update animal category by ID in the database.

**Request Body**

* Same as Add Animal

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message

### Delete Animal Category by ID

Bash: DELETE /deleteAnimalCat/:id

Delete animal category by ID from the database.

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message

### Add System

Bash: POST /addSystem

Add a new system to the database.

**Request Body**

* **sid** (string): System ID
* **sysname** (string): System name
* **password** (string): System password
* **location** (string): System location
* **type** (string): System type
* **cid** (string): System CID

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message

### Login System

Bash: POST /loginSystem

Login a system.

**Request Body**

* **sysname** (string): System name
* **password** (string): System password

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message if login successful, error message otherwise

# Authentication API Documentation

This API provides endpoints for user registration and login.

## Base URL

Arduino: <http://your-base-url.com/api/auth>

## Endpoints

### Register User

Arduino: POST /register

Register a new user.

**Request Body**

* **username** (string): User's username
* **password** (string): User's password

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message

### Login User

Bash: POST /login

Login a user.

**Request Body**

* **username** (string): User's username
* **password** (string): User's password

**Response**

* Status: 200 OK
* Content-Type: application/json
* Body: Success message with authentication token