API Documentation for FreshLens

Introduction

FreshLens is a robust application that combines TensorFlow models and external API calls to identify and provide detailed information about various types of fruits. It uses FastAPI for image processing and prediction, and Express.js for fetching fruit data from an external API.

FastAPI Endpoint Documentation

Base URL for FastAPI: `http://localhost:8000`

1. POST /predict

Description: This endpoint accepts an uploaded image to predict the type of fruit it contains using a pre-trained TensorFlow model.

Request:

- **Method:** POST
- **URL:** `/predict`
- **Body:** `multipart/form-data` which includes:
  - **file:** Image file (type: UploadFile)

Response:

- **Content-Type:** `application/json`
- **Body:** JSON object containing the identified fruit.
  - **fruit:** (string) - Predicted fruit name.

Status Codes:

- **200 OK** - Successfully predicted the fruit in the image.
- **500 Internal Server Error** - An error occurred while processing the request.

Example cURL Request:

```
curl -X POST "<http://localhost:8000/predict>" \\\\
    -F "file=./apple.jpg"
```

Express.js Endpoint Documentation

Base URL for Express.js: `http://localhost:3000`

1. GET /api/fruit/:fruitName

Description: Retrieves detailed information about a specific fruit by name from the external FruityVice API.

Request:

- **Method:** GET
- **URL:** `/api/fruit/:fruitName`
  - **fruitName:** (string) - The name of the fruit to retrieve details for.

Response:

- **Content-Type:** `application/json`
- **Body:** JSON object containing detailed information about the fruit, if found.

Status Codes:

- **200 OK** - Successfully retrieved the fruit information.
- **404 Not Found** - No fruit matches the provided name.
- **500 Internal Server Error** - Failed to retrieve fruit data.

Example cURL Request:

curl "<http://localhost:3000/api/fruit/apple>"

General Error Handling:

Both applications are equipped with error handling mechanisms. The FastAPI app uses an exception handler to return a 500 status code with a relevant message if any exceptions occur. Similarly, the Express.js application includes try-catch blocks to handle and respond to errors effectively during external API requests.