Image Processing System API Documentation

Overview

This API service processes image data from CSV files, compresses images, and provides status updates for processing requests.

Base URL

http://localhost:3000/api

API Endpoints

1. Upload CSV

Uploads a CSV file containing product and image information.

Endpoint: POST /upload

Request:

- Content-Type: multipart/form-data
- Body Parameter:
    - **file**: CSV file (required)

CSV Format:

CSV

```
1S. No.,Product Name,Input Image Urls
21,SKU1,https://example.com/image1.jpg
32,SKU2,https://example.com/image2.jpg
```

Success Response:

JSON

```
{
  "requestId": "661e656f-e896-4844-9c3a-a6650f1244c7",
  "message": "File uploaded successfully. Image processing started.",
  "productsProcessed": 2
}
```

- Status Code: 202 Accepted

Error Response:

JSON

```
{
  "error": "No file uploaded",
  "message": "Please select a CSV file to upload"
}
```

- Status Code: 400 Bad Request

2. Check Status

Retrieves the processing status of a specific request.

Endpoint: GET /status/{requestId}

Parameters:

- **requestId**: Unique identifier received from upload response (required)

Success Response:

```
{
         "requestId": "661e656f-e896-4844-9c3a-a6650f1244c7",
          "status": "COMPLETED",
         "createdAt": "2024-01-20T10:30:00.000Z",
        "updatedAt": "2024-01-20T10:31:00.000Z",
        "products": [
                    {
         "serialNumber": 1,
        "productName": "SKU1",
        "inputImageUrls": ["https://example.com/image1.jpg"],
         "outputImageUrls": ["http://localhost:3000/processed/image1_processed.jpg"]
    }
  ]
}
```

- Status Code: 200 OK

Error Response:

JSON

```
{
   "error": "Request not found",
  "requestId": "invalid-id"
}
```

- Status Code: 404 Not Found

3. Download Processed CSV

Downloads a CSV file containing both input and output image URLs.

Endpoint: GET /status/{requestId}/csv

Parameters:

- **requestId**: Unique identifier received from upload response (required)

Success Response:

- Content-Type: text/csv
- File Download: output_{requestId}.csv

CSV Output Format:

CSV

```
1S. No.,Product Name,Input Image Urls,Output Image Urls
21,SKU1,https://example.com/image1.jpg,http://localhost:3000/processed/image1_processed.jpg
```

Status Codes

- 200: Success
- 202: Accepted (Processing Started)
- 400: Bad Request
- 404: Not Found
- 500: Internal Server Error

Processing States

- PENDING: Request received
- PROCESSING: Images being processed
- COMPLETED: All processing finished

- FAILED: Processing error occurred

Image Processing

- Images are compressed to 50% of original quality
- Processed images are stored in the /uploads/processed directory
- Output URLs are generated for each processed image

Sample Usage

Using cURL

BASH

```
# Upload CSV

curl -X POST -F "file=@test.csv" http://localhost:3000/api/upload

# Check Status

curl http://localhost:3000/api/status/661e656f-e896-4844-9c3a-a6650f1244c7

# Download Processed CSV

curl http://localhost:3000/api/status/661e656f-e896-4844-9c3a-a6650f1244c7/csv
```

Using Postman

1. Upload CSV:
    - Method: POST
    - URL: http://localhost:3000/api/upload
    - Body: form-data
    - Key: file (Type: File)
    - Value: Select CSV file
2. Check Status:
    - Method: GET
    - URL: http://localhost:3000/api/status/{requestId}
3. Download CSV:
    - Method: GET
    - URL: http://localhost:3000/api/status/{requestId}/csv

Error Handling

Common error responses include:

JSON

```
{
   "error": "Error type",
   "message": "Detailed error message"
}
```