Image Processing App

Prepared by: Gurpreet Singh
Mail: gurpreetchahal8792@gmail.com
Contact No.: +91 6005114557

Table of Contents

API Documentation & Asynchronous Worker Documentation

1. Introduction

This document provides a detailed explanation of the API endpoints and the asynchronous worker process involved in the Image Processing System. The system is designed to handle image processing tasks asynchronously and notify users upon completion through a webhook mechanism.

2. API Documentation

2.1 Overview

The API allows users to upload a CSV file containing image URLs, which are then processed asynchronously. The processed images are stored, and their URLs are provided in a downloadable CSV file. Users can check the status of their processing request and receive notifications upon completion.

2.2 API Endpoints

2.2.1 Upload CSV File

This endpoint is responsible for receiving a CSV file that contains image URLs. Users can optionally provide a webhook URL where they will receive a notification once the processing is complete. The system validates the CSV format and stores the request information in the database. The request is then queued for asynchronous processing.

2.2.2 Get Processing Status

This endpoint allows users to check the current status of their request. It provides information about whether the request is still being processed or has been completed. If completed, it returns the processed image URLs along with the original input URLs.

2.2.3 Webhook Notification

A webhook notification is automatically triggered once all images for a request have been processed. This ensures that external systems can be informed in real-time when the processing is complete. The webhook receives a structured payload containing the request ID, processing status, and a link to the output CSV file containing the processed images.

2.2.4 Serve Processed Files

The system also serves processed image files and output CSV files through predefined static file routes. Users can access their processed images directly and download the final CSV file containing the results.

3. Asynchronous Workers Documentation

3.1 Overview

The image processing system relies on an asynchronous worker that runs in the background to process uploaded images. The worker operates at scheduled intervals, retrieving pending requests and executing processing tasks.

3.2 Workflow

1. The worker scans the database for image processing requests marked as 'Pending.'
2. Each image from the request is downloaded and processed (e.g., compression, resizing).
3. The processed images are stored in a designated directory, and their URLs are updated in the database.
4. If all images in a request have been processed, the system generates an output CSV file containing both the input and output image URLs.
5. If a webhook URL was provided, a notification is sent to that URL, informing the requester that processing has been completed.
6. The status of the request is updated to 'Completed.'

3.3 Webhook Mechanism

The webhook mechanism ensures that external systems can be notified when the processing task is finished. If a webhook URL is associated with a request, a structured HTTP request is sent to that URL containing the details of the processed images.

3.4 Error Handling

To ensure the reliability of the system, error handling mechanisms are in place:

- If an image download fails, it is retried a limited number of times before marking it as 'Failed.'
- If an image cannot be processed, the error is logged, and processing continues for the remaining images.
- If the webhook fails to respond, it is retried multiple times before being marked as 'Failed.'

3.5 Scalability Considerations

To enhance performance and scalability, the system can be extended using:

- Parallel processing to handle multiple image requests concurrently.
- Cloud-based storage solutions such as AWS S3 for managing large volumes of images.
- A job queue system (e.g., Redis Queue) to distribute processing tasks across multiple worker instances.

4. Low-Level Design (LLD)

4.1 System Components

The system consists of the following key components:

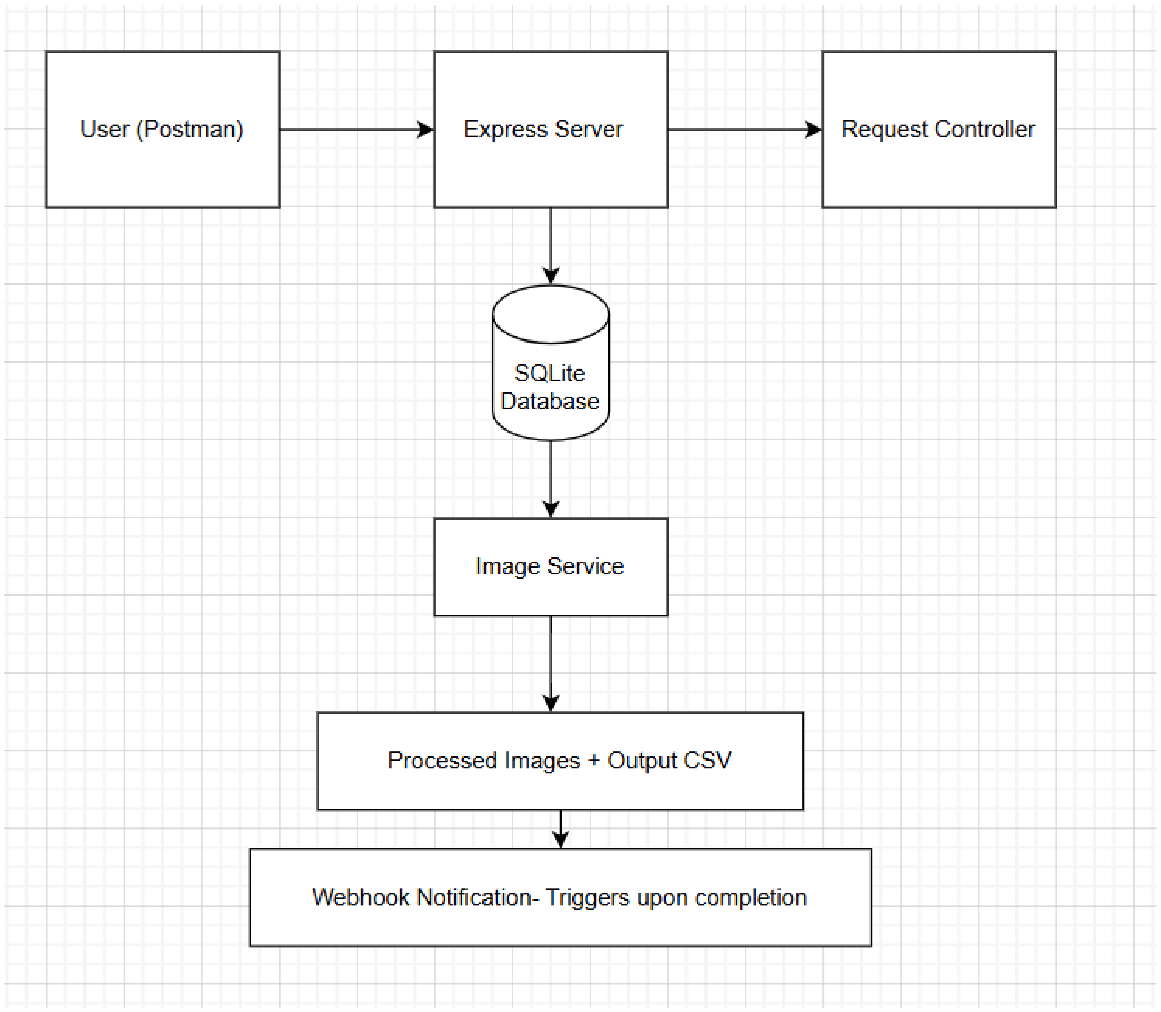
- API Gateway: Manages incoming requests and routes them to the appropriate service.
- Database: Stores request information, image processing status, and webhook URLs.
- Asynchronous Worker: Handles image processing tasks and triggers webhook notifications.
- Storage System: Manages uploaded CSV files, processed images, and output CSV files.
- Webhook Service: Sends notifications to external systems upon completion of processing.

4.2 Data Flow

1. The user uploads a CSV file through the API.
2. The API stores request details in the database and queues the request for processing.
3. The worker retrieves pending requests and processes each image.
4. The processed images are stored, and their details are updated in the database.
5. An output CSV file is generated containing input and output image URLs.
6. A webhook notification is sent if applicable.

7. Users can check the processing status or directly access the processed files.

4.3 System Diagram


User (Postman)
Express Server
Request Controller
SQLite Database
Image Service
Processed Images + Output CSV
Webhook Notification- Triggers upon completion