**Asynchronous Image Processing Service**

**Overview**

This project provides an asynchronous service to process image files (JPG/PNG) and extract text from them.  
It is built with **FastAPI**, **RabbitMQ**, and **SQLite**, wrapped in **Docker** for easy setup.

* **Producer**: FastAPI app for submitting images and querying job status/results.
* **Consumer**: Worker service that consumes jobs from RabbitMQ, extracts text, and stores results in the database.
* **Common**: Shared utilities (DB models, config, RabbitMq and Enums).

**How to Run**

1. **Clone & navigate into the project**
2. git clone <your-repo-url>
3. create the .env file in root folder
4. cd image2text
5. **Build and start services**
6. docker compose up --build

This will start:

* + RabbitMQ (UI at <http://localhost:15672>, user/pass = guest/guest)
  + Producer API (FastAPI) on <http://localhost:8000>
  + Consumer worker

1. docker compose logs -f consumer
2. docker compose logs -f producer

**API Endpoints**

**1. Submit a new image job**

**POST** /submit  
Upload a JPG/PNG image for text extraction.

* **Input**: multipart/form-data with file field = file
* **Response**: JSON containing job\_id

**2. Get job status**

**GET** /status/{job\_id}  
Check if the job is queued, processing, done, or failed.

**3. Get job result**

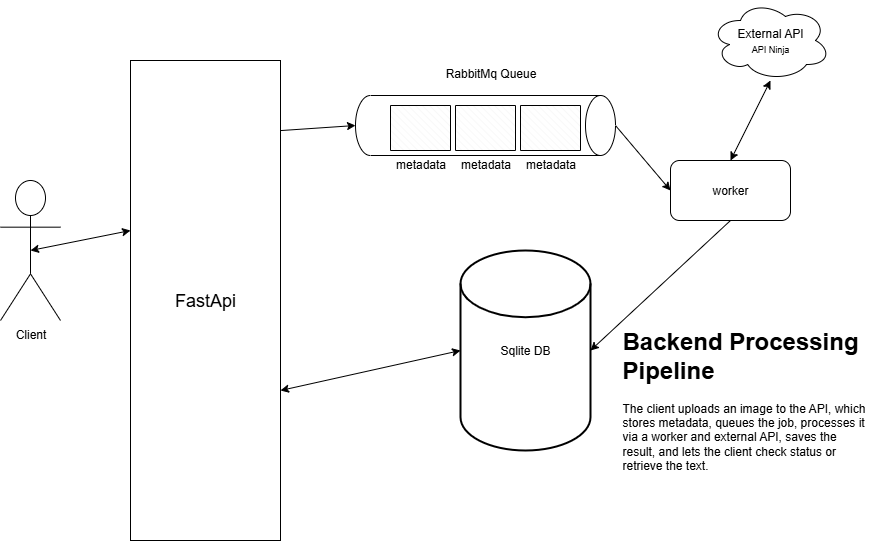
**GET** /result/{job\_id}  
If job is done, returns extracted text.

**Testing the API**

You can test the API using the included **Python client**:

cd tests

python python\_client.py

I added some sample images in tests/sample\_images/

Example test:   
A black screen with white text

AI-generated content may be incorrect.