PDF rendering service

In Rossum, we are processing hundreds of thousands documents every month. The very first step is input file normalization and rendering, so that subsequent processing may be performed. Right now, rendering is part of the data extraction process and we want to extract it to a separate service.

Assignment

Create a service that accepts PDF files containing one or more pages. These pages should be rendered to normalized png files: they should fit into 1200x1600 pixels rectangle.

The service is accessible through a REST API and offloads all file processing to asynchronous tasks (e.g. using dramatiq library), so that it is easy to scale.

REST API endpoints:

- POST /documents
  - uploads a file
  - returns JSON { “id”: “<DOCUMENT_ID>? }
- GET /documents/<DOCUMENT_ID>
  - returns JSON { “status”: “processing|done”, “n_pages”: NUMBER }
- GET /documents/<DOCUMENT_ID>/pages/<NUMBER>
  - return rendered image png

Implementation

- Python 3, Flask/Django, possibly database (e.g. sqlite, PostgreSQL), message queue (e.g. rabbitmq), possibly dramatiq or similar library
- Should include a simple Dockerfile and docker-compose.yml to enable easy testing

Notes

- API should use common practices, e.g. return proper status codes, content-type, etc.
- Consider software development best practices