Document processing: where data finds purpose.

As a software engineer at *RDI*. You are building an API for a document processing service. Users can upload images and PDF files to the API, and the API will perform some operations on the files and return the results.

Requirements:

1. Create a Django project and a REST API using Django Rest Framework (DRF).
2. The API should have the following endpoints:
    - **`POST /api/upload/`**: Accepts image and PDF files in base64 format and saves them to the server.
    - **`GET /api/images/`**: Returns a list of all uploaded images.
    - **`GET /api/pdfs/`**: Returns a list of all uploaded PDFs.
    - **`GET /api/images/{id}/`**: Returns the details of a specific image, like the location, width, height, number of channels.
    - **`GET /api/pdfs/{id}/`**: Returns the details of a specific PDF, like the location, number of pages, page width, page height.
    - **`DELETE /api/images/{id}/`**: Deletes a specific image.
    - **`DELETE /api/pdfs/{id}/`**: Deletes a specific PDF.
    - **`POST /api/rotate/`**: Accepts an image ID and rotation angle, rotates the image, and returns the rotated image.
    - **`POST /api/convert-pdf-to-image/`**: Accepts a PDF ID, converts the PDF to an image, and returns the image.
3. Implement the necessary models, serializers, views, and URLs to handle the API endpoints.
4. The image and PDF files should be saved in a directory on the server, and the file paths should be stored in the database.
5. Implement proper error handling and validation for the API endpoints.
6. Use Git for version control and commit your code regularly with meaningful commit messages.
7. Dockerize the Django project for easy deployment and testing.

Deliverables:

1. A Git repository url containing the source code of the task.
2. The code should have all the endpoints working as expected.
3. The code should contain a Dockerfile that after building will produce a working docker image, the docker image should behave like the normal application when run.

Extra Credit:

You'll gain extra credit for doing some or all of the following:

1. Add tests to the project using pytest.
2. Deploy the project on any free hosting, like pythonanywhere.
3. Provide a postman collection with examples of how to use the API.

We Like:

- ***Working Code!.***
- Clean Code.
- Descriptive commit messages.
- Unit/Integration Tests.