# **Introduction**

In this Codelab, you will learn how to build a Streamlit-based frontend for interacting with a document-processing API. This API can process documents using different techniques such as PyMuPDF, BeautifulSoup, LXML, Microsoft Document Intelligence, and PyTesseract.

By the end of this tutorial, you will be able to:

Upload PDFs, images, and DOCX files

Process documents using an API

Display extracted content in Streamlit

Download processed files from S3 storage

# **Prerequisites**

Before you start, make sure you have the following:

Python (>=3.8)

Installed dependencies:

streamlit boto3

fastapi

# **Running the Application**

For the backend create an Endpoint for each of the services using Fastapi. For ex PyMuPdf, Doc Intelligence etc.

Once the Backend Endpoints are created, make sure that you route the endpoints in a main.py file.

run the fastapi endpoints..

uvicorn main:app --reload

Then run the Streamlit Application using

Make sure the respective credentials are in env file and do call them wherever required

### 

### 

### **Note Do not Commit the env file as it will cause issues.**

Streamlit run Frontendnew.py

**Running the application**

Select the respective Dropdowns (e.x PymuPdf,Tesseract etc).

**Workflow**We have used “pymupdf” and “pytesseract” as the open-source pdf parsers, pymupdf generates a markdown file from the pdf file we upload and gets saved to the amazon S3 bucket, pytesseract takes the pdf and converts it to an “ocr file” which is also saved to the S3 bucket.

We have used “Beautiful Soap” and “LXML” as the webpage parsers, where the webpage content is extracted as images, text and tables, and all of this is saved to amazon S3 where the name of the folder will be the file which we upload.

We have used “streamlit” as the frontend resource, where we have created a main.py file that holds the addresses of the individual functions and gets the individual parsers, and the user can call the parsers he or she desires.

At the we have deployed the entire application via Amazon AWS.