# Document Parsing

A Document Parsing system

## Official Website (work in progress)

https://axatechlab.github.io/AXA-AEL-pdfparser/

## API

To start the API server, just run:

```
npm run start:api
```

The documentation is here.

## Binary dependencies for Linux and Mac OS X

We use `qpdf`, `mupdf-tools`, `imagemagick` and `pdf2json` to do process pdf files, extract fonts and convert pdf to json structure. You must install this tools on your machine prior to use docparser.

```
pacman -S qpdf mupdf-tools pdf2json imagemagick   # Arch Linux
apt-get install qpdf pdf2json imagemagick         # Debian based linux distro
```

On OS X:

```
brew install qpdf mupdf-tools pdf2json imagemagick
```

### Tesseract

https://github.com/tesseract-ocr/tesseract/

*Only used if you give an image to the pipeline.*

### Duckling

Follow this guide: https://github.com/facebook/duckling#duckling-

## Dependencies (Windows)

We recommand using Chocolatey to install dependencies. It makes things much more easier to manage.

**1) Install Chocolatey**

**2) Install available dependencies using Powershell (Run as Administrator):**

```
choco install qpdf mupdf imagemagick
```

**3) Install Node.js**

Download and install Node.js: https://nodejs.org/en/download

**4) Install pdf2json**

Download the latest release (`.msi` file) of pdf2json here: https://github.com/flexpaper/pdf2json/releases

Then, you need to add `pdf2json.exe` to your PATH.
If you have install it in `C:\Program Files (x86)\PDF2JSON`, you can either add it using the user interface or execute the following command in Powershell (Run as Administrator):

```
setx PATH "$env:PATH;C:\Program Files (x86)\PDF2JSON" -m
```

**5) Install Tesseract**

*Only required if you upload images (`jpg`, `png`, `tiff`, etc.) to the tool.*

You can download Tesseract 4.0 64-bit for Windows or check out other available format on the wiki

Then, you need to add tesseract.exe to your PATH.
If you have install it in `C:\Program Files (x86)\Tesseract-OCR`, you can either add it using the user interface execute the following command in Powershell (Run as Administrator):

```
setx PATH "$env:PATH;C:\Program Files (x86)\Tesseract-OCR" -m
```

**Duckling**

Duckling seems to be super complicated to install on Windows. To avoid this, we're working to provide it through a docker image.

**ABBYY FineReader Server Configuration**

If the ABBYY FineReader Server is to be used as the OCR extraction solution, the following environment variables need to be set on the host running `pdfparser`:

1. `ABBYY_SERVER_URL` : The network address of the ABBYY FineReader Server.
2. `ABBYY_SERVER_VER` : The major version number of the ABBYY FineReader Server. For example: 14 for ABBYY FineReader Server 14.01
3. `ABBYY_WORKFLOW` : The name of the server's workflow to be called to process the file.

On the side of the ABBYY FineReader Server, make sure the XML output is configured for the selected workflow: 1. Double click on the workflow to be used. 2. In the tab titled 'output', make sure the list of file formats exported contains the XML format; if not, add it with the 'New' button. 3. Make sure the following settings are enabled on the XML format's settings: 1. Character Attributes 2. Extended Character Attributes 3. Coordinates of the Original Image 4. Character Formatting

## Install npm dependencies

```
npm install
```

## Compile and Run

You can run the extractor from the command line:

```
# Mac OS X, Linux:
npm run run:debug -- --input-file ~/Downloads/sample1.xml --output-folder dist/ --document-r
# Windows:
cmd /C "npm run run:debug -- --input-file samples/t1.pdf --output-folder samples --document-
```

Start the Web Interface:

```
node run start:web
```

## Use

Open localhost:3000 with your favorite browser.

You can also use it with the API.

## Test

```
npm run test
```

## Developer Documentation

Project directory structure:

```
.
  README.md
  client
      index.js
      localStorage.js
      public
          css
          fonts
          img
          js
          views
      style
           style.scss
  doc
      API.md
      architecture.md
  package-lock.json
  package.json
  server
      api
      bin
          index.ts
      defaultConfig.json
      src
           assets
           cleaning-tools
           definition-finder
           duckling
           exporters
           extractor-tools
           labeling-tool
           pipelineModules
           types
  test
      assets
          text-order-mini.pdf
      helpers.ts
```

```
    line-merge.spec.ts
    number-correction.spec.ts
    paragraph-merge.spec.ts
    text-order-detection.spec.ts
    utils.spec.ts
tsconfig.json
yarn.lock
```

## CONTRIBUTE

Please refer to the guidelines in CONTRIBUTING.md.