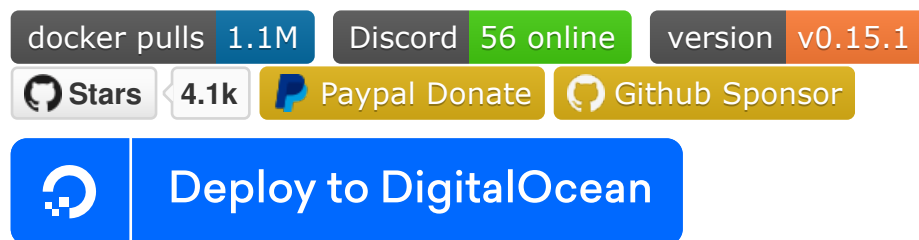


Stirling-PDF



This is a powerful locally hosted web based PDF manipulation tool using docker that allows you to perform various operations on PDF files, such as splitting merging, converting, reorganizing, adding images, rotating, compressing, and more. This locally hosted web application started as a 100% ChatGPT-made application and has evolved to include a wide range of features to handle all your PDF needs.

Stirling PDF makes no outbound calls for any record keeping or tracking.

All files and PDFs are either purely client side, in server memory only during the execution of the task or within a temporary file only for execution of the task. Any file which has been downloaded by the user will have already been deleted from the server by that time.

Feel free to request any features or bug fixes either in github issues or our Discord  
stirling-home

## Features

- Dark mode support.
- Custom download options (see here for example)
- Parallel file processing and downloads
- API for integration with external scripts
- Optional Login and Authentication support (see here for documentation)

## PDF Features

### Page Operations

- View and modify PDFs - View multi page PDFs with custom viewing sorting and searching. Plus on page edit features like annotate, draw and adding text and images. (Using PDF.js with Joxit and Liberation.Liberation fonts)
- Full interactive GUI for merging/splitting/rotating/moving PDFs and their pages.
- Merge multiple PDFs together into a single resultant file.
- Split PDFs into multiple files at specified page numbers or extract all pages as individual files.

- Reorganize PDF pages into different orders.
- Rotate PDFs in 90-degree increments.
- Remove pages.
- Multi-page layout (Format PDFs into a multi-paged page).
- Scale page contents size by set %.
- Adjust Contrast.
- Crop PDF.
- Auto Split PDF (With physically scanned page dividers).
- Extract page(s).
- Convert PDF to a single page.

### **Conversion Operations**

- Convert PDFs to and from images.
- Convert any common file to PDF (using LibreOffice).
- Convert PDF to Word/Powerpoint/Others (using LibreOffice).
- Convert HTML to PDF.
- URL to PDF.
- Markdown to PDF.

### **Security & Permissions**

- Add and remove passwords.
- Change/set PDF Permissions.
- Add watermark(s).
- Certify/sign PDFs.
- Sanitize PDFs.
- Auto-redact text.

### **Other Operations**

- Add/Generate/Write signatures.
- Repair PDFs.
- Detect and remove blank pages.
- Compare 2 PDFs and show differences in text.
- Add images to PDFs.
- Compress PDFs to decrease their filesize (Using OCRMyPDF).
- Extract images from PDF.
- Extract images from Scans.
- Add page numbers.
- Auto rename file by detecting PDF header text.
- OCR on PDF (Using OCRMyPDF).
- PDF/A conversion (Using OCRMyPDF).
- Edit metadata.
- Flatten PDFs.
- Get all information on a PDF to view or export as JSON.

For a overview of the tasks and the technology each uses please view Endpoint-groups.md Hosted instance/demo of the app can be seen here hosted by the team at adminforge.de

## Technologies used

- Spring Boot + Thymeleaf
- PDFBox
- LibreOffice for advanced conversions
- OcrMyPdf
- HTML, CSS, JavaScript
- Docker
- PDF.js
- PDF-LIB.js

## How to use

### Locally

Please view <https://github.com/Frooodle/Stirling-PDF/blob/main/LocalRunGuide.md>

### Docker

<https://hub.docker.com/r/frooodle/s-pdf>

Stirling PDF has 3 different versions, a Full version, Lite, and ultra-Lite. Depending on the types of features you use you may want a smaller image to save on space. To see what the different versions offer please look at our version mapping For people that don't mind about space optimization just use the latest tag.

Stirling-PDF Full	604 MB	Stirling-PDF Lite	420 MB
Stirling-PDF Ultra-Lite	213 MB		

Docker Run

```
docker run -d \  
  -p 8080:8080 \  
  -v /location/of/trainingData:/usr/share/tesseract-ocr/4.00/tessdata \  
  -v /location/of/extraConfigs:/configs \  
  -e DOCKER_ENABLE_SECURITY=false \  
  --name stirring-pdf \  
  frooodle/s-pdf:latest
```

Can also add these for customisation but are not required

```
-v /location/of/customFiles:/customFiles \  

```

Docker Compose

```
version: '3.3'
services:
  stirring-pdf:
    image: frooodle/s-pdf:latest
    ports:
      - '8080:8080'
    volumes:
      - /location/of/trainingData:/usr/share/tesseract-ocr/4.00/tessdata #Required for extra
      - /location/of/extraConfigs:/configs
#      - /location/of/customFiles:/customFiles/
    environment:
      - DOCKER_ENABLE_SECURITY=false
```

## Enable OCR/Compression feature

Please view <https://github.com/Frooodle/Stirling-PDF/blob/main/HowToUseOCR.md>

## Want to add your own language?

Stirling PDF currently supports 19! - English (English) (en\_GB) - English (US) (en\_US) - Arabic ( ) (ar\_AR) - German (Deutsch) (de\_DE) - French (Français) (fr\_FR) - Spanish (Español) (es\_ES) - Chinese ( ) (zh\_CN) - Catalan (Català) (ca\_CA) - Italian (Italiano) (it\_IT) - Swedish (Svenska) (sv\_SE) - Polish (Polski) (pl\_PL) - Romanian (Română) (ro\_RO) - Korean ( ) (ko\_KR) - Portuguese Brazilian (Português) (pt\_BR) - Russian ( ) (ru\_RU) - Basque (Euskara) (eu\_ES) - Japanese ( ) (ja\_JP) - Dutch (Nederlands) (nl\_NL) - Greek (el\_GR)

If you want to add your own language to Stirling-PDF please refer <https://github.com/Frooodle/Stirling-PDF/blob/main/HowToAddNewLanguage.md>

And please create a PR to merge it back in so others can use it!

## How to View

1. Open a web browser and navigate to **`http://localhost:8080/`**
2. Use the application by following the instructions on the website.

## Customisation

Stirling PDF allows easy customization of the app. Includes things like - Custom application name - Custom slogans, icons, images, and even custom HTML (via file overrides)

There are two options for this, either using the generated settings file **`settings.yml`** This file is located in the **`/configs`** directory and follows

standard YAML formatting

Environment variables are also supported and would override the settings file  
For example in the settings.yml you have

```
system:
  defaultLocale: 'en-US'
```

To have this via an environment variable you would have `SYSTEM_DEFAULTLOCALE`

The Current list of settings is

```
security:
  enableLogin: false # set to 'true' to enable login
  csrfDisabled: true
```

```
system:
  defaultLocale: 'en-US' # Set the default language (e.g. 'de-DE', 'fr-FR', etc)
  googlevisibility: false # 'true' to allow Google visibility (via robots.txt), 'false' to c
  customStaticFilePath: '/customFiles/static/' # Directory path for custom static files
```

```
#ui:
#  appName: exampleAppName # Application's visible name
#  homeDescription: I am a description # Short description or tagline shown on homepage.
#  appNameNavbar: navbarName # Name displayed on the navigation bar
```

```
endpoints:
  toRemove: [] # List endpoints to disable (e.g. ['img-to-pdf', 'remove-pages'])
  groupsToRemove: [] # List groups to disable (e.g. ['LibreOffice'])
```

```
metrics:
  enabled: true # 'true' to enable Info APIs endpoints (view http://localhost:8080/swagger-u
```

### Extra notes

- Endpoints. Currently, the endpoints `ENDPOINTS_TO_REMOVE` and `GROUPS_TO_REMOVE` can include comma separate lists of endpoints and groups to disable as example `ENDPOINTS_TO_REMOVE=img-to-pdf,remove-pages` would disable both image-to-pdf and remove pages, `GROUPS_TO_REMOVE=LibreOffice` Would disable all things that use LibreOffice. You can see a list of all endpoints and groups here
- `customStaticFilePath`. Customise static files such as the app logo by placing files in the `/customFiles/static/` directory. An example of customising app logo is placing a `/customFiles/static/favicon.svg` to override current SVG. This can be used to change any images/icons/css/fonts/js etc in Stirling-PDF

## Environment only parameters

- `SYSTEM_ROOTURIPATH` is set to `/pdf-app` to Set the application's root URI to `localhost:8080/pdf-app`
- `SYSTEM_CONNECTIONTIMEOUTMINUTES` to set custom connection timeout values
- `DOCKER_ENABLE_SECURITY` to tell docker to download security jar (required as true for auth login)

## API

For those wanting to use Stirling-PDFs backend API to link with their own custom scripting to edit PDFs you can view all existing API documentation [here](#) or navigate to `/swagger-ui/index.html` of your `stirling-pdf` instance for your versions documentation (Or by following the API button in your settings of Stirling-PDF)

## Login authentication

`stirling-login` ### Prerequisites: - User must have the folder `./configs` volumed within docker so that it is retained during updates. - Docker uses must download the security jar version by setting `DOCKER_ENABLE_SECURITY` to `true` in environment variables. - Then either enable login via the `settings.yml` file or via setting `SECURITY_ENABLE_LOGIN` to `true` - Now the initial user will be generated with username `admin` and password `stirling`. On login you will be forced to change the password to a new one. You can also use the environment variables `SECURITY_INITIALLOGIN_USERNAME` and `SECURITY_INITIALLOGIN_PASSWORD` to set your own straight away (Recommended to remove them after user creation).

Once the above has been done, on restart, a new `stirling-pdf-DB.mv.db` will show if everything worked.

When you login to Stirling PDF you will be redirected to `/login` page to login with those default credentials. After login everything should function as normal

To access your account settings go to Account settings in the settings cog menu (top right in navbar) This Account settings menu is also where you find your API key.

To add new users go to the bottom of Account settings and hit 'Admin Settings', here you can add new users. The different roles mentioned within this are for rate limiting. This is a Work in progress which will be expanding on more in future

For API usage you must provide a header with 'X-API-Key' and the associated API key for that user.

## FAQ

### Q1: What are your planned features?

- Progress bar/Tracking
- Full custom logic pipelines to combine multiple operations together.
- Folder support with auto scanning to perform operations on
- Redact text (Via UI not just automated way)
- Add Forms
- Multi page layout (Stich PDF pages together) support x rows y columns and custom page sizing
- Fill forms manual and automatic

### Q2: Why is my application downloading .htm files?

This is a issue caused commonly by your NGINX congifuration. The default file upload size for NGINX is 1MB, you need to add the following in your Nginx sites-available file. `client_max_body_size SIZE;` Where “SIZE” is 50M for example for 50MB files.

### Q3: Why is my download timing out

NGINX has timeout values by default so if you are running Stirling-PDF behind NGINX you may need to set a timeout value such as adding the config `proxy_read_timeout 3600;`