



MkDocs Exporter  
5.3.1

# Setting up PDF documents

# Setting up PDF documents

## Prerequisites

Under-the-hood, this library depends on a web browser controlled by *Playwright* to generate PDF documents.

At the time of writing, *Playwright* supports the following operating systems:

- Python 3.8 or higher
- Windows 10+, Windows Server 2016+ or Windows Subsystem for Linux (WSL)
- macOS 12 Monterey or MacOS 13 Ventura
- Debian 11, Debian 12, Ubuntu 20.04 or Ubuntu 22.04

### Your operating system is not supported?

You can still use *Docker* to build your documentation from any operating system.  
Feel free to check out the [Dockerfile](#) used by this documentation.

To install the browser and its required dependencies, run:

```
playwright install --with-deps
```

## Configuration

First of all, you'll need to register the `exporter-pdf` plugin (**after** the `exporter` one) to your configuration:

```
plugins:  
  - exporter  
  - exporter-pdf
```

You can find the configuration reference [here](#).

### Why multiple plugins?

**MkDocs Exporter** comes as various plugins in a single package.

This architecture reduces code duplication and maintains a generic base that can be used to export your pages to formats other than PDF (although this is currently the only format supported).

To sum things up, the `exporter` plugin should always be registered first as it provides a common ground for other plugins to rely on.

# Usage

## Toggling documents generation

Document generation can be enabled or disabled at any time. This feature is particularly useful during your development processes: when you don't want to slow down your iterations because of document generation.

```
plugins:  
  - exporter  
  - exporter-pdf:  
    enabled: ![MKDOCS_EXPORTER_ENABLED, true]
```

With this configuration, the `MKDOCS_EXPORTER_ENABLED` environment variable can be used as a switch for the generation process.

## Setting up cover pages

Cover pages can give your PDF documents a professional quality. Here's how cover pages are set up for this documentation.

### Power! Unlimited Power!

The following examples use the `macros` plugin for *extra powerful* pages.

`mkdocs.yml`

```
plugins:  
  - exporter  
  - exporter-pdf:  
    stylesheets:  
      - resources/stylesheets/pdf.scss  
    covers:  
      front: resources/templates/covers/front.html.j2  
      back: resources/templates/covers/back.html.j2
```

[...]

 See the full content of this file [here](#).

#### resources/templates/covers/front.html.j2

```
<div class="front-cover">
  
  <section>
    <div class="brand">{{ config.site_name }}</div>
    <div class="title">{{ page.title }}</div>
  </section>
</div>

[...]
```

🔗 See the full content of this file [here](#).

#### resources/templates/covers/back.html.j2

```
<div class="back-cover">
  <section>
    <div class="title">{{ config.site_name }}</div>
  </section>
</div>

[...]
```

🔗 See the full content of this file [here](#).

#### resources/stylesheets/pdf.scss

```
@page {
  size: A4;
  margin: 1.20cm;
}

[...]
```

🔗 See the full content of this file [here](#).

## Increasing concurrency

PDF are, by default, generated concurrently which greatly reduces the overall build time. You may want to override the default value of **4**, depending on your hardware.

```
plugins:
  - exporter-pdf:
      concurrency: 16
```

With this configuration, up to **16** PDF documents will be generated concurrently. As you've guessed, a value of **1** will force PDF documents to be built sequentially.

## Excluding some pages

You may want to prevent some pages from generating a PDF document. To do so, you can use the `pdf` meta tag on your pages:

```
---
pdf: false
---

# Lorem ipsum dolor sit amet

[...]
```

If you exclude more pages than you include, you may want to explicitly define the pages for which PDF documents should be generated.

This behaviour is called the `explicit` mode, it can be enabled in your configuration file:

```
plugins:
  - exporter-pdf:
      explicit: true
```

With this option, only pages with a truthful `pdf` meta tag will have their corresponding PDF document generated.

```
---
pdf: true
---

# Lorem ipsum dolor sit amet

[...]
```

# MkDocs Exporter