Options for MkDocs PDF Generate Plugin

Options

Revision 0.2

Duodu Randy

Disclaimer: Content can change at anytime and best to refer to website for latest information.

Table of Contents

1. Options

The plugin allows users to pass in both global and local options. Back to Home

> **Note**
>
> Local options has a higher precedence than global options.

1.1 Global Options

Some of these global options are used as default options when local options are not set. The global options are passed to the plugin through `mkdocs.yml`:

```yaml
plugins:
    - pdf-generate:
        #author: "Randy Duodu"
        #copyright: "Copyright © 2022 - Ghana"
        #disclaimer: "Disclaimer: Content can change at anytime and best to refer to website for latest information."
        #toc_level: 3
        #cover: true
        #cover_title: TITLE TEXT
        #cover_subtitle: SUBTITLE TEXT
        #custom_template_path: TEMPLATES PATH
        #toc_title: TOC TITLE TEXT
        #cover_logo: LINK TO COVER LOGO IMAGE
        #document_cover:
          #document_type: "Link to image file to use as cover logo for a specific type of documents"
          #manual: "https://resources.breadboardmates.com/img/manuals-banner.svg"
        #verbose: true
        #media_type: print
        #enabled_if_env: ENABLE_PDF_EXPORT
```

1.1.1 for Cover

`cover`

Set the value to `false` if you don't need a cover page.

default: `true`

`cover_title`

Set the title text in cover page. **NB:** We use the value for the title option in your markdown page's metadata if set.

default: use `site_name` in your project `mkdocs.yml`

`cover_subtitle`

Set the subtitle text in cover page.

default: use the value for the top-level section (`h1`) on the page

`cover_logo`

Set the logo image in cover page. This value is URL or simply specify the relative path to the documentation directory. **NB:** If a user sets the document_cover, we use the link value as the cover page.
default: `None`

`author`

Set the author text.

default: use `site_author` in your project `mkdocs.yml`

`copyright`

Set the copyright text.

default: use `copyright` in your project `mkdocs.yml`

`disclaimer`

Set the disclaimer text.

default: use `disclaimer` in your project `mkdocs.yml`

`copyright`

Set the author text.

default: use `copyright` in your project `mkdocs.yml`

`document_cover`

Set the cover_logo for specific document types. The option takes a `key-value` pair where the `key` must be the same value as the value set for type in the local options. The `value` must be the path to the image.

default: `None`

1.1.2 for Heading and TOC

`toc_title`

Set the title text of *Table of Content*.

default: `Table of Content`

`toc_level`

Set the level of *Table of Content*. This value is enabled in the range of from `1` to `3`.

default: `3`

1.1.3 ... and more

`custom_template_path`

The path where your custom `cover.html` and/or `custom.css` are located.
default: `templates`

`enabled_if_env`

Setting this option will enable the build only if there is an environment variable set to 1. This is useful to disable building the PDF files during development, since it can take a long time to export all files.

default: `None`

PDF generation can take significantly longer than HTML generation which can slow down MkDocs built-in dev-server.

Adding `enabled_if_env: ENABLE_PDF_EXPORT` disables PDF generation during development. Run the dev-server normally:

```
$ mkdocs serve
INFO     -  Building documentation...
INFO     -  PDF export is disabled (set environment variable ENABLE_PDF_EXPORT to 1 to enable)
INFO     -  Cleaning site directory
INFO     -  Documentation built in 0.54 seconds
INFO     -  [08:51:24] Watching paths for changes: 'docs', 'mkdocs.yml'
INFO     -  [08:51:24] Serving on http://127.0.0.1:8000/
```

and to build PDF files, set the `ENABLE_PDF_EXPORT=1` environment variable at the command line:

```
$ ENABLE_PDF_EXPORT=1 mkdocs build
...
INFO    -  Converting 2 files to PDF took 1.82s
INFO    -  Documentation built in 2.29 seconds
```

`verbose`

Setting this to `true` will show all WeasyPrint debug messages during the build.
default: `false`

`media_type`

Allows you to use a different CSS media type (or a custom one like `pdf-generate`) for the PDF export.
default: `print`

`theme_handler_path`

Allows you to specify a custom theme handler module. This path must be ***relative to your project root*** (See example below).
default: `Not Set`

`mkdocs.yml`:

```
plugins:
    - pdf-export:
        theme_handler_path: theme-handler.py
```

```
project-root

├── theme-handler.py

├── docs

├── mkdocs.yml

├── site

.

.
```

1.2 Local Options

The plugin allows you to set document specific options using the Markdown page metadata. If a page metadata is specified, it has higher precedence than the global options.

The local options are specified in the specific Markdown document:

```
---
pdf:
  - filename: Plugin Options
  - title: Options for MkDocs PDF Generate Plugin
  - type: Manual
  - revision: 0.1
---
```

The following options are available:

- title
- type
- filename
- revision

1.2.1 `title`

Set the title text in cover page.
NB: We use the value for the cover_title global option if you don't set this option.

1.2.2 `type`

Set the document type.
NB: We use the value for this option in selecting the document's cover logo from the document_cover_logo option.

1.2.3 `filename`

Set the filename to use for a specific page when downloading the PDF document.

1.2.4 `revision`

Set the revision text in cover page.

1.3 Custom cover page

You can create a custom cover page using a jinja2 or html file.

> **Note**
>
> The custom cover page (e.g. cover.html or cover.j2) file must be stored in the same directory as custom_template_path.

Using jinja2 syntax, you can access all data from your `mkdocs.yml`. To make template creation easier, you can use `plugin_some_plugin` to access variables from plugins. E.g. use `{{ author }}` to get the author from your `mkdocs.yml` that looks like:

```
plugins:
    - pdf-generate:
        author: WHO
```

You can use custom variables `extra:` in your `mkdocs.yml` And, you can check it in the log if run with `verbose` options.

1.4 Custom stylesheet

You can create a custom stylesheet file called `custom.css`. This file can contain any user specified styles. Since your stylesheet is appended to the default ones, you can override every setting from them.

> **Note**
>
> The custom stylesheet file must be stored in the same directory as custom_template_path.