MkDocs Exporter

6.1.1

PDF

PDF

`mkdocs_exporter.formats.pdf.config.Config`

The plugin's configuration.

Source code in `mkdocs_exporter/formats/pdf/config.py`

```python
class Config(BaseConfig):
  """The plugin's configuration."""

  enabled = c.Type(bool, default=True)
  """Is the generator enabled?"""

  explicit = c.Type(bool, default=False)
  """Should pages specify explicitly that they should be rendered as PDF?"""

  concurrency = c.Type(int, default=4)
  """The maximum number of concurrent PDF generation tasks."""

  stylesheets = c.ListOfItems(c.File(exists=True), default=[])
  """A list of custom stylesheets to apply before rendering documents."""

  scripts = c.ListOfItems(c.File(exists=True), default=[])
  """A list of custom scripts to inject before rendering documents."""

  covers = c.SubConfig(CoversConfig)
  """The document's cover pages."""

  browser = c.SubConfig(BrowserConfig)
  """The browser's configuration."""

  url = c.Optional(c.Type(str))
  """The base URL that'll be prefixed to links with a relative path."""

  aggregator = c.SubConfig(AggregatorConfig)
  """The aggregator's configuration."""
```

`aggregator = c.SubConfig(AggregatorConfig)`

> The aggregator's configuration.

`browser = c.SubConfig(BrowserConfig)`

> The browser's configuration.

`concurrency = c.Type(int, default=4)`

> The maximum number of concurrent PDF generation tasks.

`covers = c.SubConfig(CoversConfig)`

> The document's cover pages.

`enabled = c.Type(bool, default=True)`

> Is the generator enabled?

`explicit = c.Type(bool, default=False)`

> Should pages specify explicitly that they should be rendered as PDF?

`scripts = c.ListOfItems(c.File(exists=True), default=[])`

> A list of custom scripts to inject before rendering documents.

`stylesheets = c.ListOfItems(c.File(exists=True), default=[])`

> A list of custom stylesheets to apply before rendering documents.

`url = c.Optional(c.Type(str))`

> The base URL that'll be prefixed to links with a relative path.

`mkdocs_exporter.formats.pdf.config.CoversConfig`

The cover's configuration.

Source code in `mkdocs_exporter/formats/pdf/config.py`

```
21  class CoversConfig(BaseConfig):
22    """The cover's configuration."""
23
24    front = c.Optional(c.File(exists=True))
25    """The front cover template location."""
26
27    back = c.Optional(c.File(exists=True))
28    """The back cover template location."""
```

`back = c.Optional(c.File(exists=True))`

> The back cover template location.

`front = c.Optional(c.File(exists=True))`

> The front cover template location.

`mkdocs_exporter.formats.pdf.config.BrowserConfig`

The browser's configuration.

Source code in `mkdocs_exporter/formats/pdf/config.py`

```
5  class BrowserConfig(BaseConfig):
6    """The browser's configuration."""
7
8    debug = c.Type(bool, default=False)
9    """Should console messages sent to the browser be logged?"""
10
11   headless = c.Type(bool, default=True)
12   """Should the browser start in headless mode?"""
13
14   timeout = c.Type(int, default=60_000)
15   """The timeout when waiting for the PDF to render."""
16
17   args = c.ListOfItems(c.Type(str), default=[])
18   """Extra arguments to pass to the browser."""
```

`args = c.ListOfItems(c.Type(str), default=[])`

> Extra arguments to pass to the browser.

`debug = c.Type(bool, default=False)`

> Should console messages sent to the browser be logged?

`headless = c.Type(bool, default=True)`

> Should the browser start in headless mode?

`timeout = c.Type(int, default=60000)`

> The timeout when waiting for the PDF to render.

`mkdocs_exporter.formats.pdf.config.AggregatorConfig`

The aggregator's configuration.

Source code in `mkdocs_exporter/formats/pdf/config.py`

```
31  class AggregatorConfig(BaseConfig):
32    """The aggregator's configuration."""
33
34    enabled = c.Type(bool, default=False)
35    """Is the aggregator enabled?"""
36
37    output = c.Type(str, default='combined.pdf')
38    """The aggregated PDF document output file path."""
39
40    metadata = c.Type(dict, default={})
41    """Some metadata to append to the PDF document."""
42
43    covers = c.Choice(['all', 'none', 'limits', 'front', 'back'], default='all')
44    """The behavior of cover pages."""
```

`covers = c.Choice(['all', 'none', 'limits', 'front', 'back'], default='all')`

> The behavior of cover pages.

`enabled = c.Type(bool, default=False)`

> Is the aggregator enabled?

`metadata = c.Type(dict, default={})`

> Some metadata to append to the PDF document.

`output = c.Type(str, default='combined.pdf')`

> The aggregated PDF document output file path.

MkDocs Exporter