

Package ‘bookdown’

April 15, 2025

Type Package

Title Authoring Books and Technical Documents with R Markdown

Version 0.43

Maintainer Yihui Xie <xie@yihui.name>

Description

Output formats and utilities for authoring books and technical documents with R Markdown.

License GPL-3

Depends R (>= 3.5.0)

Imports htmltools (>= 0.3.6), knitr (>= 1.38), rmarkdown (>= 2.14),
jquerylib, xfun (>= 0.48), tinytex (>= 0.12), yaml (>= 2.1.19)

Suggests bslib (>= 0.2.4), downlit (>= 0.4.0), htmlwidgets, jsonlite,
curl, rstudioapi, miniUI, rsconnect (>= 0.4.3), servr (>= 0.13), shiny, tibble, testit (>= 0.9), tufte, xml2, webshot,
testthat (>= 3.1.0), withr (>= 2.3.0)

URL <https://github.com/rstudio/bookdown>,
<https://pkgs.rstudio.com/bookdown/>

BugReports <https://github.com/rstudio/bookdown/issues>

SystemRequirements Pandoc (>= 1.17.2)

RoxygenNote 7.3.2

Encoding UTF-8

Config/Needs/book remotes, webshot, svglite

Config/Needs/website pkgdown, tidyverse/tidytemplate, rstudio/quillt

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Yihui Xie [aut, cre] (<<https://orcid.org/0000-0003-0645-5666>>),
Christophe Dervieux [ctb] (<<https://orcid.org/0000-0003-4474-2498>>),
JJ Allaire [ctb],
Albert Kim [ctb],

Alessandro Samuel-Rosa [ctb],
Andrzej Oles [ctb],
Atsushi Yasumoto [ctb] (<<https://orcid.org/0000-0002-8335-495X>>),
Aust Frederik [ctb] (<<https://orcid.org/0000-0003-4900-788X>>),
Bastiaan Quast [ctb],
Ben Marwick [ctb],
Chester Ismay [ctb],
Clifton Franklund [ctb],
Daniel Emaasit [ctb],
David Shuman [ctb],
Dean Attali [ctb],
Drew Tyre [ctb],
Ellis Valentiner [ctb],
Frans van Dunne [ctb],
Hadley Wickham [ctb],
Jeff Allen [ctb],
Jennifer Bryan [ctb],
Jonathan McPhers [ctb],
JooYoung Seo [ctb] (<<https://orcid.org/0000-0002-4064-6012>>),
Joyce Robbins [ctb],
Junwen Huang [ctb],
Kevin Cheung [ctb],
Kevin Ushey [ctb],
Kim Seonghyun [ctb],
Kirill Muller [ctb],
Luciano Selzer [ctb],
Matthew Lincoln [ctb],
Maximilian Held [ctb],
Michael Sachs [ctb],
Michal Bojanowski [ctb],
Nathan Werth [ctb],
Noam Ross [ctb],
Peter Hickey [ctb],
Pedro Rafael D. Marinho [ctb] (<<https://orcid.org/0000-0003-1591-8300>>),
Romain Lesur [ctb] (<<https://orcid.org/0000-0002-0721-5595>>),
Sahir Bhatnagar [ctb],
Shir Dekel [ctb] (<<https://orcid.org/0000-0003-1773-2446>>),
Steve Simpson [ctb],
Thierry Onkelinx [ctb] (<<https://orcid.org/0000-0001-8804-4216>>),
Vincent Fulco [ctb],
Yixuan Qiu [ctb],
Zhuoer Dong [ctb],
Posit Software, PBC [cph, fnd],
Bartek Szopka [ctb] (The jQuery Highlight plugin),
Zeno Rocha [cph] (clipboard.js library),
MathQuill contributors [ctb] (The MathQuill library; authors listed in
inst/resources/AUTHORS),
FriendCode Inc [cph, ctb] (The gitbook style, with modifications)

Repository CRAN

Date/Publication 2025-04-15 16:30:06 UTC

Contents

bookdown_site	3
bs4_book	4
calibre	5
clean_book	5
create_book	6
epub_book	6
fence_theorems	8
gitbook	8
html_chapters	10
html_document2	12
pdf_book	14
publish_book	15
render_book	16
serve_book	17

Index	19
--------------	-----------

bookdown_site	<i>R Markdown site generator for bookdown</i>
---------------	-----------------------------------------------

Description

Implementation of custom R Markdown site generator for bookdown.

Usage

```
bookdown_site(input, ...)
```

Arguments

input	Website directory (or the name of a file within the directory).
...	Currently unused.

bs4_book

HTML book built with bootstrap4

Description

This output format is built with **Bootstrap**, using carefully crafted features to provide a clean reading experience whether you are on a phone, tablet, or desktop. To read more about this format, see: <https://bookdown.org/yihui/bookdown/html.html#bs4-book>

Usage

```
bs4_book(
  theme = bs4_book_theme(),
  repo = NULL,
  ...,
  lib_dir = "libs",
  pandoc_args = NULL,
  extra_dependencies = NULL,
  template = "default",
  split_bib = FALSE,
  footnotes_inline = TRUE
)

bs4_book_theme(primary = "#0068D9", version = 4, ...)
```

Arguments

theme	A named list or <code>bslib:bs_theme()</code> object. The default, <code>bs4_book_theme()</code> , resets the base font size to 1rem to make reading easier and uses a primary colour with greater contrast against the background.
repo	Either link to repository where book is hosted, used to generate view source and edit buttons or a list with repository base link, default branch, subdir and icon (see "Specifying the repository" in https://bookdown.org/yihui/bookdown/html.html#bootstrap4-style).
lib_dir, pandoc_args, extra_dependencies, split_bib, ...	Passed on to <code>rmarkdown:html_document()</code> .
template	Pandoc template to use for rendering. Pass "default" to use the bookdown default template; pass a path to use a custom template. The default template should be sufficient for most use cases. For advanced user only, in case you want to develop a custom template, we highly recommend to start from the default template: https://github.com/rstudio/bookdown/blob/master/inst/templates/bs4_book.html . Otherwise, some feature may not work anymore.
footnotes_inline	By default, footnotes will be set inline and shown on hover. Set to FALSE to keep footnotes at the bottom of the page with links.
primary	Primary colour: used for links and background of footer.

version	Passed to <code>bslib::bs_theme()</code> . This should not be changed as <code>bs4_book()</code> has been designed to work with Bootstrap version 4 for now.
---------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

calibre	<i>A wrapper function to convert e-books using Calibre</i>
---------	------------------------------------------------------------

Description

This function calls the command `ebook-convert` in Calibre (<https://calibre-ebook.com>) to convert e-books.

Usage

```
calibre(input, output, options = "")
```

Arguments

input	The input filename.
output	The output filename or extension (if only an extension is provided, the output filename will be the input filename with its extension replaced by output; for example, <code>calibre('foo.epub', 'mobi')</code> generates <code>'foo.mobi'</code>).
options	A character vector of additional options to be passed to <code>ebook-convert</code> .

Value

The output filename.

clean_book	<i>Clean up the output files and directories from the book</i>
------------	----------------------------------------------------------------

Description

After a book is rendered, there will be a series of output files and directories created in the book root directory, typically including `*_files/`, `*_cache/`, `_book/`, and some HTML/LaTeX auxiliary files. These filenames depend on the book configurations. This function identifies these files and directories, and delete them if desired, so you can rebuild the book with a clean source.

Usage

```
clean_book(clean = getOption("bookdown.clean_book", FALSE))
```

Arguments

clean	Whether to delete the possible output files. If <code>FALSE</code> , simply print out a list of files/directories that should probably be deleted. You can set the global option <code>bookdown.clean_book = TRUE</code> to force this function to delete files. You are recommended to take a look at the list of files at least once before actually deleting them, i.e. run <code>clean_book(FALSE)</code> before <code>clean_book(TRUE)</code> .
-------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

create_book *Create a bookdown project*

Description

Create a bookdown project with multiple book output formats, including HTML. Choose one of two HTML book output formats:

Usage

```
create_gitbook(path)
```

```
create_bs4_book(path)
```

Arguments

path Absolute path to an empty directory in which to create the bookdown project. In the RStudio IDE, if **rstudioapi** package available, an RStudio project will be created.

Details

- Use `create_gitbook()` to use `bookdown::gitbook()` output format
- Use `create_bs4_book()` to use a `bookdown::bs4_book()` output format

The function will create a folder with file structure for a bookdown project, and example files with information on how to start.

epub_book *The EPUB e-book format*

Description

Convert a book to the EPUB format, which is an e-book format supported by many readers, such as Amazon Kindle Fire and iBooks on Apple devices.

Usage

```
epub_book(  
  fig_width = 5,  
  fig_height = 4,  
  dev = "png",  
  fig_caption = TRUE,  
  number_sections = TRUE,  
  toc = FALSE,  
  toc_depth = 3,  
)
```

```

stylesheet = NULL,
cover_image = NULL,
metadata = NULL,
chapter_level = 1,
epub_version = c("epub3", "epub", "epub2"),
md_extensions = NULL,
global_numbering = !number_sections,
pandoc_args = NULL,
template = "default"
)

```

Arguments

<code>fig_width, fig_height, dev, fig_caption</code>	Figure options (width, height, the graphical device, and whether to render figure captions).
<code>number_sections</code>	Whether to number sections.
<code>toc, toc_depth</code>	Whether to generate a table of contents, and its depth.
<code>stylesheet</code>	A character vector of paths to CSS stylesheets to be applied to the eBook.
<code>cover_image</code>	The path to a cover image.
<code>metadata</code>	The path to the EPUB metadata file.
<code>chapter_level</code>	The level by which the e-book is split into separate “chapter” files.
<code>epub_version</code>	Whether to use version 3 or 2 of EPUB. This correspond to Pandoc’s supported output format . “epub” is an alias for “epub3” since Pandoc 2.0 and “epub2” for earlier version.
<code>md_extensions</code>	A character string of Pandoc Markdown extensions.
<code>global_numbering</code>	If TRUE, number figures and tables globally throughout a document (e.g., Figure 1, Figure 2, ...). If FALSE, number them sequentially within sections (e.g., Figure 1.1, Figure 1.2, ..., Figure 5.1, Figure 5.2, ...). Note that <code>global_numbering = FALSE</code> will not work with <code>number_sections = FALSE</code> because sections are not numbered.
<code>pandoc_args</code>	A vector of additional Pandoc arguments.
<code>template</code>	Pandoc template to use for rendering. Pass “default” to use Pandoc’s built-in template; pass a path to use a custom template. The default pandoc template should be sufficient for most use cases. In case you want to develop a custom template, we highly recommend to start from the default EPUB templates at https://github.com/jgm/pandoc-templates/ .

Note

Figure/table numbers cannot be generated if sections are not numbered (`number_sections = FALSE`).

fence_theorems	<i>Convert the syntax of theorem and proof environments from code blocks to fenced Divs</i>
----------------	---------------------------------------------------------------------------------------------

Description

This function converts the syntax `````{theorem, label, ...}` to `::: {.theorem #label ...}` (Pandoc's fenced Div) for theorem environments.

Usage

```
fence_theorems(input, text = xfun::read_utf8(input), output = NULL)
```

Arguments

input	Path to an Rmd file that contains theorem environments written in the syntax of code blocks.
text	A character vector of the Rmd source. When text is provided, the input argument will be ignored.
output	The output file to write the converted input content. You can specify output to be identical to input, which means the input file will be overwritten. If you want to overwrite the input file, you are strongly recommended to put the file under version control or make a backup copy in advance.

Value

If `output = NULL`, the converted text is returned, otherwise the text is written to the output file.

References

Learn more about [theorems and proofs](#) and [custom blocks](#) in the **bookdown** book.

gitbook	<i>The GitBook output format</i>
---------	----------------------------------

Description

This output format function ported a style provided by GitBook (<https://www.gitbook.com>) for R Markdown. To read more about this format, see: <https://bookdown.org/yihui/bookdown/html.html#gitbook-style>

Usage

```
gitbook(
  fig_caption = TRUE,
  number_sections = TRUE,
  self_contained = FALSE,
  anchor_sections = TRUE,
  lib_dir = "libs",
  global_numbering = !number_sections,
  pandoc_args = NULL,
  extra_dependencies = list(),
  ...,
  template = "default",
  split_by = c("chapter", "section", "0", "1", "2", "3", "4", "5", "6", "chapter+number",
    "section+number", "0+number", "1+number", "2+number", "3+number", "4+number",
    "5+number", "6+number", "rmd", "none"),
  split_bib = TRUE,
  config = list(),
  table_css = TRUE,
  code_folding = c("none", "show", "hide")
)
```

Arguments

`fig_caption`, `number_sections`, `self_contained`, `anchor_sections`, `lib_dir`,
`pandoc_args`, `code_folding`, `extra_dependencies`, ...

Arguments to be passed to `rmarkdown::html_document()` (... not including `toc`, and `theme`).

`global_numbering`

If TRUE, number figures and tables globally throughout a document (e.g., Figure 1, Figure 2, ...). If FALSE, number them sequentially within sections (e.g., Figure 1.1, Figure 1.2, ..., Figure 5.1, Figure 5.2, ...). Note that `global_numbering = FALSE` will not work with `number_sections = FALSE` because sections are not numbered.

`template`

Pandoc template to use for rendering. Pass "default" to use the bookdown default template; pass a path to use a custom template. The default template should be sufficient for most use cases. In case you want to develop a custom template, we highly recommend to start from the default template: <https://github.com/rstudio/bookdown/blob/master/inst/templates/gitbook.html>.

`split_by`

How to name the HTML output files from the book: `rmd` uses the base filenames of the input Rmd files to create the HTML filenames, e.g. generate 'chapter1.html' for 'chapter1.Rmd'; `none` or "0" means do not split the HTML file (the book will be a single HTML file); `chapter` or "1" means split the file by the first-level headers; `section` or "2" means the second-level headers, "3"- "6" means split the file by the [3-6]-level headers. For `chapter`, `section` and "1"- "6", the HTML filenames will be determined by the header ID's, e.g. the filename for the first chapter with a chapter title # Introduction will be 'introduction.html'; for "chapter+number", "section+number" and

	"[1-6]+number" the chapter/section (and higher level section) numbers will be prepended to the HTML filenames, e.g. '1-introduction.html' and '2-1-literature.html'.
split_bib	Whether to split the bibliography onto separate pages where the citations are actually used.
config	A list of configuration options for the gitbook style, such as the font/theme settings.
table_css	TRUE to load gitbook's default CSS for tables. Choose FALSE to unload and use customized CSS (for example, bootstrap) via the css option. Default is TRUE.

html_chapters	<i>Build book chapters into separate HTML files</i>
---------------	-----------------------------------------------------

Description

Split the HTML output into chapters while updating relative links (e.g. links in TOC, footnotes, citations, figure/table cross-references, and so on). Functions `html_book()` and `tufte_html_book()` are simple wrapper functions of `html_chapter()` using a specific base output format.

Usage

```
html_chapters(
  toc = TRUE,
  number_sections = TRUE,
  fig_caption = TRUE,
  lib_dir = "libs",
  template = bookdown_file("templates/default.html"),
  global_numbering = !number_sections,
  pandoc_args = NULL,
  ...,
  base_format = rmarkdown::html_document,
  split_bib = TRUE,
  page_builder = build_chapter,
  split_by = c("chapter", "section", "0", "1", "2", "3", "4", "5", "6", "chapter+number",
    "section+number", "0+number", "1+number", "2+number", "3+number", "4+number",
    "5+number", "6+number", "rmd", "none")
)

html_book(...)

tufte_html_book(...)
```

Arguments

`toc`, `number_sections`, `fig_caption`, `lib_dir`, `template`, `pandoc_args`

See `rmarkdown::html_document`, `tufte::tufte_html`, or the documentation of the `base_format` function.

global_numbering	If TRUE, number figures and tables globally throughout a document (e.g., Figure 1, Figure 2, ...). If FALSE, number them sequentially within sections (e.g., Figure 1.1, Figure 1.2, ..., Figure 5.1, Figure 5.2, ...). Note that <code>global_numbering = FALSE</code> will not work with <code>number_sections = FALSE</code> because sections are not numbered.
...	Other arguments to be passed to <code>base_format</code> . For <code>html_book()</code> and <code>tufte_html_book()</code> , ... is passed to <code>html_chapters()</code> .
base_format	An output format function to be used as the base format.
split_bib	Whether to split the bibliography onto separate pages where the citations are actually used.
page_builder	A function to combine different parts of a chapter into a page (an HTML character vector). See build_chapter for the specification of this function.
split_by	How to name the HTML output files from the book: <code>rmd</code> uses the base filenames of the input Rmd files to create the HTML filenames, e.g. generate 'chapter1.html' for 'chapter1.Rmd'; <code>none</code> or <code>"0"</code> means do not split the HTML file (the book will be a single HTML file); <code>chapter</code> or <code>"1"</code> means split the file by the first-level headers; <code>section</code> or <code>"2"</code> means the second-level headers, <code>"3"-6</code> means split the file by the [3-6]-level headers. For <code>chapter</code> , <code>section</code> and <code>"1"-6</code> , the HTML filenames will be determined by the header ID's, e.g. the filename for the first chapter with a chapter title # Introduction will be 'introduction.html'; for "chapter+number", "section+number" and "[1-6]+number" the chapter/section (and higher level section) numbers will be prepended to the HTML filenames, e.g. '1-introduction.html' and '2-1-literature.html'.

Value

An R Markdown output format object to be passed to `bookdown::render_book()`.

Note

These functions are expected to be used in conjunction with `render_book()`. It is almost meaningless if they are used with `rmarkdown::render()`. Functions like [html_document2](#) are designed to work with the latter.

If you want to use a different template, the template must contain three pairs of HTML comments: `<!--bookdown:title:start-->` and `<!--bookdown:title:end-->` to mark the title section of the book (this section will be placed only on the first page of the rendered book); `<!--bookdown:toc:start-->` and `<!--bookdown:toc:end-->` to mark the table of contents section (it will be placed on all chapter pages); `<!--bookdown:body:start-->` and `<!--bookdown:body:end-->` to mark the HTML body of the book (the HTML body will be split into separate pages for chapters). You may open the default HTML template (`bookdown::bookdown_file('templates/default.html')`) to see where these comments were inserted.

html_document2	<i>Output formats that allow numbering and cross-referencing figures/tables/equations</i>
----------------	-------------------------------------------------------------------------------------------

Description

These are simple wrappers of the output format functions like `rmarkdown::html_document()`, and they added the capability of numbering figures/tables/equations/theorems and cross-referencing them. See ‘References’ for the syntax. Note you can also cross-reference sections by their ID’s using the same syntax when sections are numbered. In case you want to enable cross reference in other formats, use `markdown_document2` with `base_format` argument.

Usage

```
html_document2(
  ...,
  number_sections = TRUE,
  global_numbering = !number_sections,
  pandoc_args = NULL,
  base_format = rmarkdown::html_document
)

html_fragment2(..., number_sections = FALSE)

html_notebook2(..., number_sections = FALSE)

html_vignette2(..., number_sections = FALSE)

ioslides_presentation2(..., number_sections = FALSE)

slidy_presentation2(..., number_sections = FALSE)

tufte_html2(..., number_sections = FALSE)

pdf_document2(...)

beamer_presentation2(..., number_sections = FALSE)

tufte_handout2(...)

tufte_book2(...)

markdown_document2(
  number_sections = TRUE,
  fig_caption = TRUE,
  md_extensions = NULL,
  global_numbering = !number_sections,
```

```

    pandoc_args = NULL,
    ...,
    base_format = rmarkdown::md_document
)

context_document2(...)

github_document2(...)

odt_document2(...)

powerpoint_presentation2(...)

rtf_document2(...)

word_document2(...)

```

Arguments

..., fig_caption, md_extensions, pandoc_args

Arguments to be passed to a specific output format function. For a function `foo2()`, its arguments are passed to `foo()`, e.g. ... of `html_document2()` are passed to `rmarkdown::html_document()`.

number_sections

Whether to number section headers: if TRUE, figure/table numbers will be of the form `X.i`, where `X` is the current first-level section number, and `i` is an incremental number (the `i`-th figure/table); if FALSE, figures/tables will be numbered sequentially in the document from 1, 2, ..., and you cannot cross-reference section headers in this case.

global_numbering

If TRUE, number figures and tables globally throughout a document (e.g., Figure 1, Figure 2, ...). If FALSE, number them sequentially within sections (e.g., Figure 1.1, Figure 1.2, ..., Figure 5.1, Figure 5.2, ...). Note that `global_numbering = FALSE` will not work with `number_sections = FALSE` because sections are not numbered.

base_format

An output format function to be used as the base format.

Value

An R Markdown output format object to be passed to `rmarkdown::render()`.

Note

These output formats are used to generate single output files, such as a single HTML output file (unlike `gitbook`, which generates multiple HTML output files by default).

The functions `'tufte_*()'` are wrappers of functions in the **tufte** package.

References

<https://bookdown.org/yihui/bookdown/>

pdf_book

Convert R Markdown to a PDF book

Description

Convert R Markdown files to PDF after resolving the special tokens of **bookdown** (e.g., the tokens for references and labels) to native LaTeX commands.

Usage

```
pdf_book(
  toc = TRUE,
  number_sections = TRUE,
  fig_caption = TRUE,
  pandoc_args = NULL,
  ...,
  base_format = rmarkdown::pdf_document,
  toc_unnumbered = TRUE,
  toc_appendix = FALSE,
  toc_bib = FALSE,
  quote_footer = NULL,
  highlight_bw = FALSE
)
```

Arguments

<code>toc</code> , <code>number_sections</code> , <code>fig_caption</code> , <code>pandoc_args</code>	See <code>rmarkdown::pdf_document</code> , or the documentation of the <code>base_format</code> function.
<code>...</code>	Other arguments to be passed to <code>base_format</code> .
<code>base_format</code>	An output format function to be used as the base format.
<code>toc_unnumbered</code>	Whether to add unnumbered headers to the table of contents.
<code>toc_appendix</code>	Whether to add the appendix to the table of contents.
<code>toc_bib</code>	Whether to add the bibliography section to the table of contents.
<code>quote_footer</code>	If a character vector of length 2 and the quote footer starts with three dashes ('---'), <code>quote_footer[1]</code> will be prepended to the footer, and <code>quote_footer[2]</code> will be appended; if <code>NULL</code> , the quote footer will not be processed.
<code>highlight_bw</code>	Whether to convert colors for syntax highlighting to black-and-white (grayscale).

Details

This function is based on `rmarkdown::pdf_document` (by default) with better default arguments. You can also change the default format to other LaTeX/PDF format functions using the `base_format` argument.

The global R option `bookdown.post.latex` can be set to a function to post-process the LaTeX output. This function takes the character vector of the LaTeX output as its input argument, and should return a character vector to be written to the `.tex` output file. This gives you full power to post-process the LaTeX output.

Note

This output format can only be used with `render_book()`.

publish_book

Publish a book to the web

Description

Publish a book to the web. Note that you should be sure to render all versions of the book before publishing, unless you have specified `render = TRUE`.

Usage

```
publish_book(
  name = NULL,
  account = NULL,
  server = NULL,
  render = c("none", "local", "server")
)
```

Arguments

name	Name of the book (this will be used in the URL path of the published book). Defaults to the <code>book_filename</code> in <code>_bookdown.yml</code> if not specified.
account	Account name to publish to. Will default to any previously published to account or any single account already associated with server.
server	Server to publish to (by default <code>beta.rstudioconnect.com</code> but any RStudio Connect server can be published to).
render	Rendering behavior for site: <ul style="list-style-type: none"> • "none" uploads a static version of the current contents of the site directory. • "local" renders the site locally then uploads it. • "server" uploads the source of the site to render on the server.

Note that for "none" and "local" source files (e.g. `.R`, `.Rmd` and `.md`) will not be uploaded to the server.

render_book

*Render multiple R Markdown documents into a book***Description**

Render multiple R Markdown files under the current working directory into a book. It can be used in the RStudio IDE (specifically, the knit field in YAML). The `preview_chapter()` function is a wrapper of `render_book(preview = TRUE)`.

Usage

```
render_book(
  input = ".",
  output_format = NULL,
  ...,
  clean = TRUE,
  envir = parent.frame(),
  clean_envir = !interactive(),
  output_dir = NULL,
  new_session = NA,
  preview = FALSE,
  config_file = "_bookdown.yml"
)

preview_chapter(..., envir = parent.frame())
```

Arguments

input	A directory, an input filename or multiple filenames. For a directory, ‘index.Rmd’ will be used if it exists in this (book) project directory. For filenames, if <code>preview = TRUE</code> , only files specified in this argument are rendered, otherwise all R Markdown files specified by the book are rendered.
output_format, ..., clean, envir	Arguments to be passed to <code>rmarkdown::render()</code> . For <code>preview_chapter()</code> , ... is passed to <code>render_book()</code> . See <code>rmarkdown::render()</code> and the bookdown reference book for details on how output formatting options are set from YAML or parameters supplied by the user when calling <code>render_book()</code> .
clean_envir	This argument has been deprecated and will be removed in future versions of bookdown .
output_dir	The output directory. If NULL, a field named <code>output_dir</code> in the configuration file ‘_bookdown.yml’ will be used (possibly not specified, either, in which case a directory name ‘_book’ will be used).
new_session	Whether to use new R sessions to compile individual Rmd files (if not provided, the value of the <code>new_session</code> option in ‘_bookdown.yml’ is used; if this is also not provided, <code>new_session = FALSE</code>).

preview	Whether to render and preview the input files specified by the input argument. Previewing a certain chapter may save compilation time as you actively work on this chapter, but the output may not be accurate (e.g. cross-references to other chapters will not work).
config_file	The book configuration file.

Details

There are two ways to render a book from Rmd files. The default way (`new_session = FALSE`) is to merge Rmd files into a single file and render this file. You can also choose to render each individual Rmd file in a new R session (`new_session = TRUE`).

Examples

```
# see https://bookdown.org/yihui/bookdown for the full documentation
if (file.exists("index.Rmd")) bookdown::render_book("index.Rmd")
## Not run:
# will use the default format defined in index.Rmd or _output.yml
bookdown::render_book("index.Rmd")
# will use the options for format defined in YAML metadata
bookdown::render_book("index.Rmd", "bookdown::pdf_book")
# If you pass an output format object, it must have all the options set
bookdown::render_book("index.Rmd", bookdown::pdf_book(toc = FALSE))

# will render the book in the current directory
bookdown::render_book()
# this is equivalent to
bookdown::render_book("index.Rmd")
# will render the book living in the specified directory
bookdown::render_book("my_book_project")

## End(Not run)
```

serve_book	<i>Continuously preview the HTML output of a book using the servr package</i>
------------	--------------------------------------------------------------------------------------

Description

When any files are modified or added to the book directory, the book will be automatically re-compiled, and the current HTML page in the browser will be refreshed. This function is based on `servr::httpw()` to continuously watch a directory.

Usage

```
serve_book(
  dir = ".",
  output_dir = "_book",
  preview = TRUE,
```

```
    in_session = TRUE,  
    quiet = FALSE,  
    ...  
  )
```

Arguments

<code>dir</code>	The root directory of the book (containing the Rmd source files).
<code>output_dir</code>	The directory for output files; see <code>render_book()</code> .
<code>preview</code>	Whether to render the modified/added chapters only, or the whole book; see <code>render_book()</code> .
<code>in_session</code>	Whether to compile the book using the current R session, or always open a new R session to compile the book whenever changes occur in the book directory.
<code>quiet</code>	Whether to suppress output (e.g., the knitting progress) in the console.
<code>...</code>	Other arguments passed to <code>servr::http()</code> (not including the handler argument, which has been set internally).

Details

For `in_session = TRUE`, you will have access to all objects created in the book in the current R session: if you use a daemonized server (via the argument `daemon = TRUE`), you can check the objects at any time when the current R session is not busy; otherwise you will have to stop the server before you can check the objects. This can be useful when you need to interactively explore the R objects in the book. The downside of `in_session = TRUE` is that the output may be different with the book compiled from a fresh R session, because the state of the current R session may not be clean.

For `in_session = FALSE`, you do not have access to objects in the book from the current R session, but the output is more likely to be reproducible since everything is created from new R sessions. Since this function is only for previewing purposes, the cleanness of the R session may not be a big concern. You may choose `in_session = TRUE` or `FALSE` depending on your specific applications. Eventually, you should run `render_book()` from a fresh R session to generate a reliable copy of the book output.

Index

beamer_presentation2 (html_document2),
12

bookdown_site, 3

bs4_book, 4

bs4_book(), 5

bs4_book_theme (bs4_book), 4

bslib::bs_theme(), 4, 5

build_chapter, 11

calibre, 5

clean_book, 5

context_document2 (html_document2), 12

create_book, 6

create_bs4_book (create_book), 6

create_gitbook (create_book), 6

epub_book, 6

fence_theorems, 8

gitbook, 8

github_document2 (html_document2), 12

html_book (html_chapters), 10

html_chapters, 10

html_document, 9, 10, 12

html_document2, 11, 12

html_fragment2 (html_document2), 12

html_notebook2 (html_document2), 12

html_vignette2 (html_document2), 12

http, 17, 18

ioslides_presentation2
(html_document2), 12

markdown_document2 (html_document2), 12

odt_document2 (html_document2), 12

pdf_book, 14

pdf_document, 14, 15

pdf_document2 (html_document2), 12

powerpoint_presentation2
(html_document2), 12

preview_chapter (render_book), 16

publish_book, 15

render, 13, 16

render_book, 11, 15, 16, 18

rmarkdown::html_document(), 4

rtf_document2 (html_document2), 12

serve_book, 17

slidy_presentation2 (html_document2), 12

tufte_book2 (html_document2), 12

tufte_handout2 (html_document2), 12

tufte_html, 10

tufte_html2 (html_document2), 12

tufte_html_book (html_chapters), 10

word_document2 (html_document2), 12