

Package ‘bookdown’

December 21, 2018

Type Package

Title Authoring Books and Technical Documents with R Markdown

Version 0.9

Maintainer Yihui Xie <xie@yihui.name>

Description

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

License GPL-3

Imports htmltools (>= 0.3.6), knitr (>= 1.16), rmarkdown (>= 1.9),
xfun (>= 0.2), tinytex, yaml (>= 2.1.14)

Suggests htmlwidgets, rstudioapi, miniUI, rsconnect (>= 0.4.3), servr
(>= 0.3), shiny, testit (>= 0.5), tufte, webshot

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

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

SystemRequirements Pandoc (>= 1.17.2)

LazyData TRUE

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Author Yihui Xie [aut, cre] (<<https://orcid.org/0000-0003-0645-5666>>),

JJ Allaire [ctb],

Albert Kim [ctb],

Alessandro Samuel-Rosa [ctb],

Andrzej Oles [ctb],

Bastiaan Quast [ctb],

Ben Marwick [ctb],

Chester Ismay [ctb],

Christophe Dervieux [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],
 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],
 Noam Ross [ctb],
 Peter Hickey [ctb],
 Sahir Bhatnagar [ctb],
 Steve Simpson [ctb],
 Thierry Onkelinx [ctb] (<<https://orcid.org/0000-0001-8804-4216>>),
 Vincent Fulco [ctb],
 Yixuan Qiu [ctb],
 Zhuoer Dong [ctb],
 RStudio Inc [cph],
 Bartek Szopka [ctb] (The jQuery Highlight plugin),
 jQuery Foundation [cph] (jQuery library),
 jQuery contributors [ctb, cph] (jQuery library; authors listed in
 inst/resources/AUTHORS),
 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 2018-12-21 09:40:11 UTC

R topics documented:

| | |
|--------------------------|---|
| bookdown_site | 3 |
| calibre | 3 |
| clean_book | 4 |
| epub_book | 4 |
| gitbook | 5 |
| html_chapters | 6 |
| html_document2 | 7 |
| kindlegen | 9 |
| pdf_book | 9 |

| | |
|-------------------------------------|-----------|
| <code>bookdown_site</code> | 3 |
| <code>publish_book</code> | 10 |
| <code>render_book</code> | 11 |
| <code>serve_book</code> | 12 |
| Index | 13 |

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

Description

Implementation of custom R Markdown site generator for bookdown.

Usage

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

Arguments

| | |
|--------------------|---|
| <code>input</code> | Website directory (or the name of a file within the directory). |
| <code>...</code> | Currently unused. |

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

Description

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

Usage

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

Arguments

| | |
|----------------------|--|
| <code>input</code> | The input filename. |
| <code>output</code> | 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 'foo.mobi'). |
| <code>options</code> | 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 FALSE, 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> . |
|-------|--|

| | |
|-----------|-------------------------------|
| epub_book | <i>The EPUB e-book format</i> |
|-----------|-------------------------------|

Description

Convert a book to the EPUB format, which is 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"), md_extensions = NULL, pandoc_args = NULL, template = "default")
```

Arguments

| | |
|---|--|
| fig_width, fig_height, dev, fig_caption | Figure options (width, height, the graphical device, and whether to render figure captions). |
| number_sections | Whether to number sections. |
| toc, toc_depth | Whether to generate a table of contents, and its depth. |

| | |
|---------------|---|
| stylesheet | A character vector of paths to CSS stylesheets to be applied to the eBook. |
| cover_image | The path to a cover image. |
| metadata | The path to the EPUB metadata file. |
| chapter_level | The level by which the e-book is split into separate “chapter” files. |
| epub_version | Whether to use version 3 or 2 of EPUB. |
| md_extensions | A character string of Pandoc Markdown extensions. |
| pandoc_args | A vector of additional Pandoc arguments. |
| template | 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`).

 gitbook

The GitBook output format

Description

This output format function ported a style provided by GitBook (<https://www.gitbook.com>) for R Markdown.

Usage

```
gitbook(fig_caption = TRUE, number_sections = TRUE, self_contained = FALSE,
        lib_dir = "libs", pandoc_args = NULL, ..., template = "default",
        split_by = c("chapter", "chapter+number", "section", "section+number",
                    "rmd", "none"), split_bib = TRUE, config = list(), table_css = TRUE)
```

Arguments

| | |
|--|---|
| fig_caption, number_sections, self_contained, lib_dir, pandoc_args | ... Arguments to be passed to <code>rmarkdown::html_document()</code> (... not including toc, and theme). |
| ... | 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> . |
| 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 means do not split the HTML file (the book will be a single HTML file); chapter means split the file by the first-level headers; section means the second-level headers. For chapter and section, 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 and section+number, the chapter/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 exmaple, 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"),
  pandoc_args = NULL, ..., base_format = rmarkdown::html_document, split_bib = TRUE,
  page_builder = build_chapter, split_by = c("section+number", "section",
    "chapter+number", "chapter", "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.

`...` Other arguments to be passed to `base_format`. For `html_book()` and `tufte_html_book()`, `...` is passed to `html_chapters()`.

`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: rmd uses the base filenames of the input Rmd files to create the HTML filenames, e.g. generate 'chapter1.html' for 'chapter1.Rmd'; none means do not split the HTML file (the book will be a single HTML file); chapter means split the file by the first-level headers; section means the second-level headers. For chapter and section, 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 and section+number, the chapter/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.

Usage

```
html_document2(..., number_sections = TRUE, pandoc_args = NULL,  
               base_format = rmarkdown::html_document)  
  
tufte_html2(..., number_sections = FALSE)  
  
pdf_document2(...)  
  
tufte_handout2(...)  
  
tufte_book2(...)  
  
word_document2(fig_caption = TRUE, md_extensions = NULL, pandoc_args = NULL,  
               ...)
```

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.

`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 function are expected to work with a single R Markdown document instead of multiple documents of a book, so they are to be passed to `rmarkdown::render()` instead of `bookdown::render_book()`. The functions ‘`tufte_*`’ are wrappers of functions in the **tufte** package.

References

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

| | |
|-----------|--|
| kindlegen | <i>A wrapper function to convert EPUB to the Mobipocket format</i> |
|-----------|--|

Description

This function simply calls the command line tool `kindlegen` provided by Amazon to convert EPUB e-books to the Mobipocket format (`.mobi`).

Usage

```
kindlegen(epub, exec = Sys.which("kindlegen"))
```

Arguments

| | |
|-------------------|--|
| <code>epub</code> | The path to a <code>.epub</code> file (e.g. created from the <code>epub_book()</code> format). If missing, it is automatically guessed from the book configurations. |
| <code>exec</code> | The path to the executable <code>kindlegen</code> , which can be downloaded from http://www.amazon.com/gp/feature.html?ie=UTF8&docId=1000765211 . |

Value

The path of the `.mobi` file if the conversion is successful.

| | |
|-----------------------|---|
| <code>pdf_book</code> | <i>Convert R Markdown to a PDF book</i> |
|-----------------------|---|

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. |

| | |
|--------------|--|
| toc_appendix | Whether to add the appendix to the table of contents. |
| toc_bib | Whether to add the bibliography section to the table of contents. |
| quote_footer | If a character vector of length 2 and the quote footer starts with three dashes ('---'), quote_footer[1] will be prepended to the footer, and quote_footer[2] will be appended; if NULL, the quote footer will not be processed. |
| highlight_bw | 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 | <i>Publish a book to the web</i> |
|--------------|----------------------------------|

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: "none" to upload a static version of the current contents of the site directory; "local" to render the site locally then upload it; "server" to render the site on the server. Note that for "none" and "local" R scripts (.R) and markdown documents (.Rmd and .md) will not be uploaded to the server. |

| | |
|-------------|---|
| render_book | <i>Render multiple R Markdown documents into a book</i> |
|-------------|---|

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,
            encoding = "UTF-8", config_file = "_bookdown.yml")

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

Arguments

| | |
|----------------------------------|--|
| input | An input filename (or multiple 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> . |
| clean_envir | Whether to clean up the environment <code>envir</code> before rendering the book. By default, the environment is cleaned when rendering the book in a non-interactive R session. |
| output_dir | The output directory. If <code>NULL</code> , a field named <code>output_dir</code> in the configuration file <code>'_bookdown.yml'</code> will be used (possibly not specified, either, in which case a directory name <code>'_book'</code> 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 <code>'_bookdown.yml'</code> 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 <code>input</code> 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). |
| encoding | Ignored. The character encoding of all input files is supposed to be UTF-8. |
| 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")
```

| | |
|------------|--|
| 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::http()` 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

bookdown_site, [3](#)
build_chapter, [7](#)

calibre, [3](#)
clean_book, [4](#)

epub_book, [4](#), [9](#)

gitbook, [5](#)

html_book (html_chapters), [6](#)
html_chapters, [6](#)
html_document, [5–7](#)
html_document2, [7](#), [7](#)
http, [12](#)

kindlegen, [9](#)

pdf_book, [9](#)
pdf_document, [9](#), [10](#)
pdf_document2 (html_document2), [7](#)
preview_chapter (render_book), [11](#)
publish_book, [10](#)

render, [8](#), [11](#)
render_book, [7](#), [10](#), [11](#), [12](#)

serve_book, [12](#)

tufte_book2 (html_document2), [7](#)
tufte_handout2 (html_document2), [7](#)
tufte_html, [6](#)
tufte_html2 (html_document2), [7](#)
tufte_html_book (html_chapters), [6](#)

word_document2 (html_document2), [7](#)