

Package ‘bookdown’

August 29, 2016

Type Package

Title Authoring Books with R Markdown

Version 0.1

Maintainer Yihui Xie <xie@yihui.name>

Description Output formats and utilities for authoring books with R Markdown.

License GPL-3

Imports htmltools (>= 0.3.5), knitr (>= 1.13), miniUI, rmarkdown (>= 0.9.6), yaml, rsconnect (>= 0.4.3)

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

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

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

LazyData TRUE

RoxygenNote 5.0.1

NeedsCompilation no

Author Yihui Xie [aut, cre],
JJ Allaire [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 2016-07-13 10:19:19

R topics documented:

| | |
|--------------------------|----|
| bookdown_site | 2 |
| calibre | 2 |
| clean_book | 3 |
| epub_book | 3 |
| gitbook | 4 |
| html_chapters | 5 |
| html_document2 | 6 |
| kindlegen | 7 |
| pdf_book | 8 |
| publish_book | 9 |
| render_book | 9 |
| serve_book | 10 |

| | |
|--------------|-----------|
| Index | 12 |
|--------------|-----------|

| | |
|---------------|---|
| 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 |

| | |
|---------|--|
| calibre | <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

| | |
|---------|--|
| 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 'foo.mobi'). |
| 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 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 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)
```

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.

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", ..., split_by = c("chapter", "chapter+number", "section",
        "section+number", "rmd", "none"), split_bib = TRUE, config = list())
```

Arguments

| | |
|--|---|
| fig_caption, number_sections, self_contained, lib_dir, ... | Arguments to be passed to <code>rmarkdown::html_document()</code> (... not including toc, theme, and template). |
| 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> means do not split the HTML file (the book will be a single HTML file); <code>chapter</code> means split the file by the first-level headers; <code>section</code> means the second-level headers. For <code>chapter</code> and <code>section</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 <code>chapter+number</code> and <code>section+number</code> , 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. |

html_chapters

Build book chapters into separate HTML files

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"),
  ..., 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
```

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

| | |
|---------------------------|---|
| ... | 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> . |
| <code>base_format</code> | An output format function to be used as the base format. |
| <code>split_bib</code> | Whether to split the bibliography onto separate pages where the citations are actually used. |
| <code>page_builder</code> | 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. |
| <code>split_by</code> | 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> means do not split the HTML file (the book will be a single HTML file); <code>chapter</code> means split the file by the first-level headers; <code>section</code> means the second-level headers. For <code>chapter</code> and <code>section</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 <code>chapter+number</code> and <code>section+number</code> , 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</i> |
|----------------|---|

Description

These are simple wrappers of the output format functions like `rmarkdown::html_document()`, and they added the capability of numbering figures/tables and cross-referencing them. See References for the syntax. Note you can also cross-reference sections by their ID's using the same syntax as figures/tables.

Usage

```
html_document2(..., number_sections = TRUE)

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.

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

<http://rstudio.github.io/bookdown/figures.html>

kindlegen

A wrapper function to convert EPUB to the Mobipocket format

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

`epub` The path to a .epub file (e.g. created from the `epub_book()` format). If missing, it is automatically guessed from the book configurations.

`exec` The path to the executable kindlegen, 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.

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

Description

Convert R Markdown files to PDF while 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, ...,
         base_format = rmarkdown::pdf_document)
```

Arguments

`toc`, `number_sections`, `fig_caption`
See `rmarkdown::pdf_document`, or the documentation of the `base_format` function.

`...` Other arguments to be passed to `base_format`.

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

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.

| | |
|--------------|----------------------------------|
| 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,
  force_knit = FALSE, preview = FALSE, encoding = "UTF-8")
```

```
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>). |
| force_knit | Whether to force knitting all Rmd files (this argument is only for <code>new_session = TRUE</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. |

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`). In this case, Rmd files that have not been updated from the previous run will not be recompiled the next time by default, and you can force compiling them by `force_knit = TRUE`.

| | |
|------------|--|
| 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 recompiled, 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, ...)
```

Arguments

| | |
|------------|---|
| dir | The root directory of the book (containing the Rmd source files). |
| output_dir | The directory for output files; see render_book() . |
| preview | Whether to render the modified/added chapters only, or the whole book; see render_book() . |
| in_session | 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. |
| ... | 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, [2](#)
build_chapter, [6](#)

calibre, [2](#)
clean_book, [3](#)

epub_book, [3](#), [8](#)

gitbook, [4](#)

html_book (html_chapters), [5](#)
html_chapters, [5](#)
html_document, [5](#), [6](#)
html_document2, [6](#), [6](#)
http, [10](#), [11](#)

kindlegen, [7](#)

pdf_book, [8](#)
pdf_document, [8](#)
pdf_document2 (html_document2), [6](#)
preview_chapter (render_book), [9](#)
publish_book, [9](#)

render, [7](#), [10](#)
render_book, [6](#), [9](#), [11](#)

serve_book, [10](#)

tufte_book2 (html_document2), [6](#)
tufte_handout2 (html_document2), [6](#)
tufte_html, [5](#)
tufte_html2 (html_document2), [6](#)
tufte_html_book (html_chapters), [5](#)

word_document2 (html_document2), [6](#)