Package 'listdown'

April 3, 2023

Title Create R Markdown from Lists

Version 0.5.7

Description Programmatically create R Markdown documents from lists.

License Apache License (>= 2.0)

Encoding UTF-8

Depends R (>= 4.0.0)

Language en-US

Imports checkmate, rmarkdown, tibble, yaml, fs

Suggests DT, ggplot2, testthat, purrr, knitr

Enhances workflowr

URL https://github.com/kaneplusplus/listdown

BugReports https://github.com/kaneplusplus/listdown/issues

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation no

Author Michael J. Kane [aut, cph, cre]
      (<https://orcid.org/0000-0003-1899-6662>)

Maintainer Michael J. Kane <michael.kane@yale.edu>

Repository CRAN

Date/Publication 2023-04-03 00:30:02 UTC

R topics documented:

---

as_ld_yml   *Turn a Computational Component List into YAML with Class Infor-*
                                   mation

Description

Create an object of type yaml::yml from a list of computational components. The function recursively descends into the list and when an element type is not a list the class information substituted for the object.

Usage

```
as_ld_yml(x)
```

Arguments

x   a named list of computational components.

Value

A yaml string with the structure of a 'listdown' object.

Examples

```
if (require("ggplot2")) {

  cc_list <- list(
    Linear = ggplot(anscombe, aes(x = x1, y = y1)) + geom_point(),
   `Non Linear` = ggplot(anscombe, aes(x = x2, y = y2)) + geom_point(),
   `Outlier Vertical`= ggplot(anscombe, aes(x = x3, y = y3)) + geom_point(),
   `Outlier Horizontal` =  ggplot(anscombe, aes(x = x4, y = y4)) +
     geom_point())

  as_ld_yml(cc_list)
}
```

---

class_and_tag *Prepend Class Information and Add Attributes*

Description

listdown decorators map list element to functions. This function is provided for convenience to prepend a class and attributes, which can then be used by custom decorators to display those element.

Usage

```
class_and_tag(.x, new_class, ...)
```

Arguments

`.x`	an object to add class and attribute information to.
`new_class`	the name of the class to be prepended to .x.
`...`	the attributes to attach to .x.

Value

The object with updated class information.

---

create_load_cc_expr *Create an expression to load a Computational Component*

Description

An expression to load a computational component can be either a raw expression, a variable holding the expression, or a string. The return is an unevaluated expression.

Usage

```
create_load_cc_expr(load_cc_expr)
```

Arguments

`load_cc_expr`	a string or expression that should be use to load the computational components.

Value

An call object that loads a computational component when evaluated.

---

ld_build_html_site   *Build an html Site from listdown Document Bundles*

Description

This function creates an html website with each tab in the page being described by a listdown
document bundle.

Usage

```
ld_build_html_site(
  doc_bundles,
  site_yaml,
  site_dir = tempdir(),
  rmd_dir = file.path(site_dir, "rmarkdown"),
  data_dir = file.path(site_dir, "data"),
  html_dir = file.path(site_dir, "html"),
  render_site = TRUE,
  view = interactive(),
  make_data_dir = TRUE,
  make_rmd_dir = TRUE,
  ...
)
```

Arguments

doc_bundles	a named list of document bundles. There can be up to one unnamed bundle, which will be assumed to correspond to an index.rmd file.
site_yaml	a list of site information, which will be written to the _site.yml file.
site_dir	the directory where the site (rmd, data, and html files) will be written to.
rmd_dir	the directory where the R Markdown files will reside. By default an "rmark-down" file is written to 'tempdir()'.
data_dir	the location where data can be found for each bundle. If the data is held in memory for a listdown document bundle, then it will be written to the specified directory. If multiple directories are specified, then the directory is specified per bundle, with index recycling used if the number of directories is not the same as the number of bundles.
html_dir	the location of the rendered document, relative to the directory specified by 'rmd_dir'. Note that this is an rmarkdown convention. By default a directory names "html" is created in the directory specified by 'rmd_dir' and rendered documents are place there.
render_site	should the page be rendered? If not then the 'html_dir' is not created.
view	should the output document be opened after rendering? By default, if 'ren-der_doc' is 'TRUE' and this argument is 'TRUE' then the browser will open for you to examine the output.

make_data_dir	if the 'data_dir' directory is not present, should it be created? This can be set to 'FALSE' when data already resides on disk to verify that it is not being created and written.
make_rmd_dir	if the 'rmd_dir' directory is not present, should it be created? This can be set to 'FALSE' when data already resides on disk to verify that it is not being created and written.
...	argument to be passed to the 'rmarkdown::render_site()' function.

Value

The path to the created website.

See Also

ld_bundle_doc ld_create_doc

---

ld_bundle_doc	*Create a 'listdown' Document Bundle*

Description

A page bundle encapsulates the computational components, R Markdown header, and listdown object. Together, these three objects are sufficient to create a document, which can be written with the 'ld_create_document()' function.

Usage

```
ld_bundle_doc(cc, header, ld)
```

Arguments

cc	the computational component list that will be presented.
header	a 'list' with the header information for the document.
ld	a 'listdown' object describing how to present the computational components.

Value

An S3 object of type 'ld_page_bundle'.

- cc the computational components associated with the document.
- ld the 'listdown' object assoiated with the document.
- header the 'listdown_header' object associated with the document.
- cc_in_memory a logical value indicating if the computational components are currently stored in memory. If TRUE, then they will be written to the disk upon creation of the document. Otherwise, the 'load_cc_expr' in the 'ld' element should have the expression to load the components.

See Also

ld_create_document

listdown

Examples

```
library(ggplot2)
cc <- list(
    iris = iris,
     Sepal.Length = list(
       Sepal.Width = ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
             geom_point(),
       Petal.Length = ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
             geom_point(),
     Colored = list(
          Sepal.Width = ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width,
            color = Species)) + geom_point(),
          Petal.Length = ggplot(iris, aes(x = Sepal.Length, y = Petal.Length,
            color = Species)) + geom_point())))

header <- ld_rmarkdown_header("Test header", author = "Some Dude",
                             date = "2020")

ld <- listdown(package = "ggplot2")

ld_bundle_doc(cc, header, ld)
```

---

ld_cc_dendro   *Show the list of Computational Components as a Dendrogram*

Description

This function creates text dendrograms from a list of computational components. It is useful for
creating a dendrogram of the the computational components of a listdown object allowing the user
to view the components hierarchically.

Usage

```
ld_cc_dendro(x)
```

Arguments

x   a named list of computational components

Value

A string, with class 'ld_cc_dendro', that shows the hierarchical structure of the computational com-
ponents (a list).

Examples

```
if (require("ggplot2")) {

  cc_list <- list(
    Linear = ggplot(anscombe, aes(x = x1, y = y1)) + geom_point(),
   `Non Linear` = ggplot(anscombe, aes(x = x2, y = y2)) + geom_point(),
   `Outlier Vertical`= ggplot(anscombe, aes(x = x3, y = y3)) + geom_point(),
   `Outlier Horizontal` =  ggplot(anscombe, aes(x = x4, y = y4)) +
     geom_point())

  ld_cc_dendro(cc_list)
}
```

---

ld_chunk_opts   *Apply Chunk Options to a Presentation Object*

Description

This function allows the user to set chunk options for individual elements of a presentation list.

Usage

```
ld_chunk_opts(pres_obj, chunk_name = NULL, ..., chunk_opts = NULL)
```

Arguments

pres_obj	the presentation list element whose chunk options should be modified.
chunk_name	the name of the chunk. By default this is NULL, corresponding to no chunk name.
...	named chunk options and their values.
chunk_opts	list of chunk options can be specified. Takes priority over arguments provided to ...

Value

An updated version of the 'pres_obj' object, with updated chunk options.

---

ld_create_doc   *Create a Document from a 'listdown' Bundle*

Description

This function creates a document, defined by a listdown bundle in a specified location on disk and, optionally, opens the document in the browser.

Usage

```
ld_create_doc(
  ldb,
  rmd_file_name = basename(tempfile(pattern = "rmarkdown", fileext = ".Rmd")),
  rmd_dir = tempdir(),
  output_dir = rmd_dir,
  render_doc = TRUE,
  cc_file_name = NULL,
  data_dir = ".",
  view = interactive(),
  ...
)
```

Arguments

ldb	a listdown doc bundle.
rmd_file_name	the name of the R Markdown file to create. By default, a temporary file is created.
rmd_dir	the directory where the output R Markdown file should be written to. By default, this is 'tempdir()'.
output_dir	the location of the rendered document, relative to the directory specified by 'rmd_dir'. Note that this is an rmarkdown convention. By default a directory names "pres" is created in the directory specified by 'rmd_dir' and rendered documents are place there.
render_doc	should the page be rendered? If not then the 'output_dir' is not created.
cc_file_name	the name of the list specifying the computational components. If this is 'NULL' (the default) then the listdown bundle is checked to make sure it's 'load_cc_expr' attribute has been specified. If it is specified, and the bundles 'load_cc_expr' has not been specified, then it will be written to disk (in the corresponding data directory, specified by 'data_dir') and read via the 'saveRDS()' function.
data_dir	the directory where data should be written. If the 'cc_file_name' argument is 'NULL' then this argument is ignored. If the 'cc_file_name' argument is specified but 'data_dir' is not, then 'tempdir()' is used.
view	should the output document be opened after rendering? By default, if 'render_doc' is 'TRUE' and this argument is 'TRUE' then the browser will open for you to examine the output.
...	options to send to the rmarkdown::render() function.

Value

The 'ldb' object passed (as 'invisible()').

See Also

ld_bundle_doc

---

ld_make_chunks *Write a listdown Object to a String*

Description

After a presentation list and listdown object have been constructed the chunks can be rendered to a string, which can be appended to a file, with appropriate headers, resulting in a compilable R Markdown document.

Usage

```
ld_make_chunks(ld, rmd_dir)
```

Arguments

ld   the listdown object that provides information on how a presentation object should
            be displayed in the output.

rmd_dir   the R Markdown directory.

Value

The string containing the R Markdown content.

See Also

listdown

---

ld_rmarkdown_header   *Create an R Markdown Header*

Description

Output an R Markdown header with specified parameters.

Usage

```
ld_rmarkdown_header(
  title,
  author = NULL,
  date = NULL,
  output = c("html_document", "pdf_document", "word_document")
)
```

Arguments

title	the title of the page.
author	the author of the page. The default is NULL - no author.
date	the date for the page. The default is NULL - no date.
output	the output format of the page. If NULL then no output format. The default is an html document.

Value

An S3 instance of class 'listdown_header' containing the R Markdown header data.

---

ld_site_yaml   *Create a Minimal Site YAML List*

Description

Create a Minimal Site YAML List

Usage

```
ld_site_yaml(site_name, tab_name, rmd_name, navbar_title = site_name)
```

Arguments

site_name	the name of the site.
tab_name	the name of the tabs on the site.
rmd_name	the name of the R Markdown files that will generate the respective tabs.
navbar_title	the title of the navigation bar (Default is the 'site_name' argument.

Value

A list with yaml components for creating a site with multiple tabs.

---

ld_workflowr_header   *Create a workflowr Header*

Description

Output a workflowr R Markdown header with specified title.

Usage

```
ld_workflowr_header(title, toc = FALSE)
```

Arguments

title	the title of the page.
toc	should the table of contents be generated? Default FALSE.

Value

An S3 instance of class 'listdown_header' containing the R Markdown 'workflowr' header data.

---

ld_write_file   *Write to an R Markdown File*

Description

This function takes header information and a listdown object and writes to a specified file.

Usage

```
ld_write_file(rmd_header, ld, file_name)
```

Arguments

rmd_header	either a character or listdown_header with R Markdown header information.
ld	the listdown object that provides information on how a presentation object should be displayed in the output.
file_name	the output file to write to.

Value

The invisible value 'TRUE' indicating the write was successful.

---

listdown   *Create a listdown Object*

Description

A listdown object provides information for how a presentation list should be used to create an R Markdown document. It requires an unquoted expression indicating how the presentation list will be loaded. In addition, libraries required by the outputted document and other parameters can be specified.

Usage

```
listdown(
  package = NULL,
  decorator = list(),
  decorator_chunk_opts = list(),
  default_decorator = identity,
  setup_expr = NULL,
  init_expr = NULL,
  load_cc_expr = NULL,
  ...,
  chunk_opts = NULL
)
```

Arguments

package
: a quoted list of package required by the outputted document.

decorator
: a named list mapping the potential types of list elements to a decorator function.

decorator_chunk_opts
: a named list mapping the potential types of list elements to chunk options that should be included for those types.

default_decorator
: the decorator to use for list elements whose type is not inherited from the decorator list. If NULL then the those elements will not be included when the chunks are written. By default this is identity, meaning that the elements will be passed directly (through the identity() function).

setup_expr
: an expression that is added before package are loaded. The expression is put into a chunk named 'setup' with option 'include = FALSE' and is intended for initializing the document. For example the expression 'knitr::opts_chunk$set(echo = FALSE)' could be used to turn echo'ing off for the entire document.

init_expr
: an initial expression that will be added to the outputted document after the libraries have been called. This expression appears after packages are loaded and before data is read.

load_cc_expr
: either an unquoted expression or a character string that will be turned into an unquoted expression via str2lang to load the presentation list.

`...`	default options sent to the chunks of the outputted document.
`chunk_opts`	a named list of options sent to the chunks of outputted documents. Note: takes priority over argument provided to ...

Value

An S3 object of type 'listdown'

- load_cc_expr - the R expression to load computational components.
- decorator - the list element decorators.
- package - package dependencies for the document.
- init_expr - the R expression to run initially.
- setup_expr - the R expression to run on setup.
- decorator_chunk_options - the R Markdown chunk options.
- default_decorator - the set of default decorators.
- chunk_opts - the default R Markdown chunk options.

Examples

```
library(ggplot2)
cc <- list(
    iris = iris,
     Sepal.Length = list(
       Sepal.Width = ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
            geom_point(),
       Petal.Length = ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
            geom_point(),
     Colored = list(
         Sepal.Width = ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width,
           color = Species)) + geom_point(),
         Petal.Length = ggplot(iris, aes(x = Sepal.Length, y = Petal.Length,
           color = Species)) + geom_point())))

header <- ld_rmarkdown_header("Test header", author = "Some Dude",
                                 date = "2020")

ld <- listdown(package = "ggplot2")

ld_bundle_doc(cc, header, ld)
```

Index