

Package: quire (via r-universe)

September 30, 2026

Title A Word-Like Report Builder for Shiny

Version 0.2.17

Description Quire's report builder in a Shiny app: a Word-like editor of documents and slide decks that writes Word, PowerPoint and PDF files itself, in the browser. The app is its host: it says what charts and tables it can draw and draws them (with 'ggplot2' or any other way), and keeps the reports; everything else, the editing, the page layout, the files and an AI's reading and editing of reports, is done by the builder.

License file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1)

Imports htmltools (>= 0.5.9), jsonlite (>= 2.0.0), shiny (>= 1.14.0)

Suggests chromote (>= 0.5.1), pdftools (>= 3.9.0), ggplot2 (>= 4.0.3), magick (>= 2.9.1), rsvg (>= 2.7.0), svglite (>= 2.2.2), testthat (>= 3.3.2), V8 (>= 8.2.0)

Config/testthat/edition 3

Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev

Repository <https://damurka.r-universe.dev>

Date/Publication 2026-09-30 01:27:10 UTC

RemoteUrl <https://github.com/damurka/quire>

RemoteRef HEAD

RemoteSha 5ce5302333dc62c3d6dbd93e9b45876ee7f9159c

Contents

quire_can_print	2
quire_chart_options	2

quire_converter	3
quire_dependency	3
quire_dispatch	4
quire_error	4
quire_export	5
quire_file_store	6
quire_host	6
quire_html_pdf	8
quire_kind	8
quire_plot	9
quire_print_pages	10
quire_server	10
quire_table	11
quire_theme_from_file	11
quire_to_pdf	12
quire_ui	12
quire_word_finish	13
Index	14

quire_can_print	<i>Whether this computer can print the builder's printable page</i>
-----------------	---

Description

Whether this computer can print the builder's printable page

Usage

quire_can_print()

Value

TRUE when the chromote package and a Chromium browser (Chrome, Edge, Chromium) are there: then [quire_html_pdf\(\)](#) and [quire_print_pages\(\)](#) work.

quire_chart_options	<i>The chart options every Quire host shares</i>
---------------------	--

Description

The one description of the chart options (from Quire's contract, chart-options.json): each option's type and, for a choice, its values; and the chart panel that edits them – its elements, its settings with their labels in each language, and its texts. The report builder has the same file, so a host that draws charts validates and lists the options from here instead of keeping its own copy.

Usage

```
quire_chart_options()
```

Value

A list: options (by name: type, and choices for a choice), tabs (the panel's elements: key, label, icon, and show, the option that shows or hides it), fields (its settings: key, tab, group, label, type, and choices, keywords, step, min, max when they have them) and texts (the panel's own words). Labels are lists by language (en, fr, pt).

quire_converter	<i>The program that turns a Word or PowerPoint file into a PDF</i>
-----------------	--

Description

Microsoft Word or PowerPoint (Windows), else LibreOffice. With one of them the PDF has exactly the file's pages.

Usage

```
quire_converter(kind = c("document", "deck"))
```

Arguments

kind "document" (a Word file) or "deck" (a PowerPoint file).

Value

"word" or "powerpoint", "libreoffice", or NULL when none is installed.

quire_dependency	<i>Quire's scripts and styles</i>
------------------	-----------------------------------

Description

The report builder's browser bundle (quire.js), its styles and fonts, as an HTML dependency. [quire_ui\(\)](#) includes it; call it yourself only to load the builder in a page of your own.

Usage

```
quire_dependency()
```

Value

An `htmltools::htmlDependency()`.

quire_dispatch	<i>Answer one message from the report builder</i>
----------------	---

Description

Calls the host's function for a call from the builder and makes its answer. `quire_server()` does this for every message; it is exported for tests and for hosts that carry the messages themselves.

Usage

```
quire_dispatch(host, message)
```

Arguments

host	A host made with <code>quire_host()</code> (or a list of functions named as its arguments).
message	A message from the builder, as a list: kind, id, method, args.

Value

The result to send back (a list: kind = "result", id, ok, and result or error), or NULL when the message is not a call.

quire_error	<i>A block the host could not draw</i>
-------------	--

Description

A block the host could not draw

Usage

```
quire_error(message)
```

Arguments

message	Why, for the reader.
---------	----------------------

Value

An error drawing, shown in the block's place.

quire_export*Write a report as a Word, PowerPoint, PDF or printable file, from R*

Description

The same writers as the builder's downloads (Quire's), run in R through 'V8', with no browser: a report written here is the file the builder would have made. The host draws the report's charts and tables as it does for the builder (its render), gives its fields and pictures; the pictures are turned and cropped with 'magick', SVG drawings made pictures with 'rsvg'.

Usage

```
quire_export(  
  project,  
  host,  
  file,  
  format = c("docx", "pptx", "pdf", "html"),  
  lang = NULL,  
  fields = NULL,  
  pdf = NULL  
)
```

Arguments

project	The report (a list, as saved), or its id (the host's getReport finds it).
host	The host, made with quire_host() .
file	Where to write it.
format	"docx" (Word; a slide deck is written as PowerPoint), "pptx", "pdf" or "html" (the printable page).
lang	The language of the report's texts when it has none of its own.
fields	The fields' values; by default the host's fields.
pdf	For a PDF: function(input, output) that converts the Word (or PowerPoint) file input to the PDF output (Word or LibreOffice).

Value

file, invisibly.

quire_file_store	<i>Keep the reports and their pictures in a folder</i>
------------------	--

Description

Makes the host's `listReports`, `getReport`, `saveReport`, `deleteReport`, `assetGet` and `assetSet`, keeping each report as a JSON file in `dir` (its pictures in `dir/assets`), as the builder sends it. Give them to `quire_host()` with `do.call()` or `...`.

Usage

```
quire_file_store(dir)
```

Arguments

<code>dir</code>	The folder (made if it is not there).
------------------	---------------------------------------

Value

A named list of functions.

quire_host	<i>Describe the app as the report builder's host</i>
------------	--

Description

The host answers the builder's few questions. Only `kinds` and `render` are needed; each other one adds a feature (the text fields, report templates, keeping reports in the app, pictures, extra ribbon groups...). Without `listReports`/`getReport`/`saveReport` the reports are kept in the reader's browser.

Usage

```
quire_host(
  kinds,
  render,
  fields = NULL,
  fieldCatalog = NULL,
  presets = NULL,
  preset = NULL,
  themes = NULL,
  years = NULL,
  regions = NULL,
  flag = NULL,
  listReports = NULL,
  getReport = NULL,
```

```

    saveReport = NULL,
    deleteReport = NULL,
    assetGet = NULL,
    assetSet = NULL,
    saveFile = NULL,
    extensions = NULL,
    action = NULL,
    ...
)

```

Arguments

kinds	function(): the charts and tables the app can draw, a list of kinds, each a list with kind, type ("chart" or "table"), label, group and groupLabel (see quire_kind()).
render	function(request): draws one block. request\$block is the block (its kind, and settings such as year, region, options), request\$width and request\$height its size in inches, request\$lang the report's language. Returns quire_plot() , quire_table() or quire_error() .
fields	function(project, lang): the values of the text fields ({country}, {report_date}...), a named list.
fieldCatalog	function(): the fields that can be inserted, a list of list(key, label).
presets, preset	function(lang): the report templates; function(id, lang, name): one as a new report.
themes	function(): the report themes.
years, regions	function(): the years and regions in the data (the charts' Year and Region lists).
flag	function(): the country's flag, as a data URL, or NULL.
listReports, getReport, saveReport, deleteReport	Keeping reports in the app: function() the saved reports (list(id, name, updated, kind) each), function(id) one report, function(project) save one, function(id) delete one. quire_file_store() makes them from a folder.
assetGet, assetSet	Pictures kept once in the app: function(id), function(id, asset).
saveFile	function(file): a Word, PowerPoint or PDF file the builder made (file\$name, file\$type, file\$data in base 64). Without it the reader's browser downloads the file.
extensions	function(): the app's own ribbon groups, fields and actions.
action	function(id, context): one of the app's own actions was pressed.
...	Any other method of the host protocol, by name.

Value

A host, to give [quire_server\(\)](#).

Other methods

pdf = function(html, name) makes a PDF of the printable page (html), returned as list(type = "application/pdf", d
pages = function(html, name) makes its pages as they are printed, for Print Preview: list(pages = <list of image dat
A host that gives neither gets Quire's own (quire_html_pdf(), quire_print_pages()), which
print with Chrome, Edge or Chromium when the computer has one (and chromote, pdftools); with-
out them the builder prints in the reader's browser.

quire_html_pdf	<i>The builder's printable page as a PDF</i>
----------------	--

Description

Prints the page (the HTML the builder makes a PDF from) with a Chromium browser, its page size, margins, headers and footers as the page sets them (CSS @page). Needs the chromote package and Chrome, Edge or Chromium.

Usage

quire_html_pdf(html, pdf)

Arguments

html	The printable page: its HTML, or an .html file.
pdf	The PDF to write.

Value

pdf, invisibly.

quire_kind	<i>A kind of chart or table the app draws</i>
------------	---

Description

A kind of chart or table the app draws

Usage

```
quire_kind(  
  kind,  
  type = c("chart", "table"),  
  label,  
  group = "charts",  
  groupLabel = group,  
  ...  
)
```

Arguments

kind	Its id.
type	"chart" or "table".
label	Its name, a string or a list of names by language (<code>list(en = "...", fr = "...")</code>).
group, groupLabel	The group it is listed in (its id and name).
...	Anything else a kind may say: <code>year = TRUE</code> (it has a year), <code>regional = TRUE</code> , <code>indicators</code> , <code>levels</code> , <code>variants</code> (lists of <code>list(value, label)</code>), <code>defaults</code> (the block's settings when inserted).

Value

A kind, a list.

quire_plot

A chart drawn for the report builder

Description

Draws a plot at the size the builder asks for. With 'svglite' installed the drawing is an SVG (sharp at any zoom, and made a picture in the files by the builder); without it, a PNG.

Usage

```
quire_plot(plot, request, chart = NULL, dpi = 144)
```

Arguments

plot	A 'ggplot2' plot, or a function that draws on the current device.
request	The render request (its width and height, in inches).
chart	Optionally, the chart as data (<code>list(type, categories, series, ...)</code>): Word and PowerPoint then have it as a real chart, its data editable there.
dpi	The PNG's dots per inch (without 'svglite').

Value

A drawing, to return from the host's render.

quire_print_pages	<i>The pages of the builder's printable page, as pictures</i>
-------------------	---

Description

Prints the page to PDF ([quire_html_pdf\(\)](#)) and renders each page as a picture: what the builder's Print Preview shows. Needs the pdftools package too.

Usage

```
quire_print_pages(html, dpi = 80)
```

Arguments

html	The printable page: its HTML, or an .html file.
dpi	Resolution of the pictures.

Value

list(pages = <list of PNG data URLs>, converter = "browser").

quire_server	<i>The report builder's host in a Shiny app's server</i>
--------------	--

Description

Answers the builder's calls with the host's functions, in the module id.

Usage

```
quire_server(id, host, on_event = NULL)
```

Arguments

id	The module's id (as given to quire_ui()).
host	The host, made with quire_host() .
on_event	Optionally, function(name, data), told what happens in the builder: "ai.ask" (the reader asked the AI: data\$prompt, data\$about), "report.opened" (data\$project, an id or NULL), "lang.changed".

Value

A list of functions to tell the builder things: send(name, data) sends an event ("data.changed" when the app's data changed, so the charts are drawn again; "report.changed" with list(project = ...) when a report changed elsewhere; "reports.changed" when the list of reports did; "report.open" with list(id = ...) to open one; "lang" with list(lang = ...), dataChanged() as a shortcut.

quire_table	<i>A table drawn for the report builder</i>
-------------	---

Description

The builder draws the table itself, in the report's style and its Table Design; the host gives its cells.

Usage

```
quire_table(data, footer = NULL, widths = NULL, text = NULL)
```

Arguments

data	A data frame: its column names are the header, its rows the table's rows (numbers kept as values, shown as <code>format()</code> gives them unless text says otherwise).
footer	A note under the table (its source), or NULL.
widths	The columns' shares of the width (summing to 1), or NULL for equal columns.
text	Optionally, a data frame of the same shape with the cells' texts.

Value

A table, to return from the host's render.

quire_theme_from_file	<i>A report theme from a PowerPoint or Word file</i>
-----------------------	--

Description

Reads the colours and fonts of an Office theme or template (PowerPoint .pptx / .potx, Word .docx / .dotx) into the fields of a report design, with the report builder's own reader (Quire's, run here through 'V8'), so a theme read in R is the one the builder reads: the accent is the theme's first accent colour, the text colour its first dark colour, the chart palette its six accent colours, the heading and body fonts its heading and body fonts. A Word file's styles give the heading colour (Heading 1's), the fonts and the text sizes when they set them, and its page size, orientation and margins. A PowerPoint file's master gives the title colour and fonts, the slide size and the background colour, and its slides their designs: the logos, bands and title boxes they carry.

Usage

```
quire_theme_from_file(path, name = NULL, default_accent = NULL)
```

Arguments

path	The file.
name	The theme's name (by default the file's name without its extension).
default_accent	The accent when the file's theme has none.

Value

A list of the design fields the file sets: `template_kind` ("pptx" or "docx"), `theme`, `name` (`list(en, fr, pt)`), `accent`, `heading_color`, `text_color`, `muted_color`, `note_fill`, `note_border`, `apply_palette`, and when the file has them `heading_font`, `body_font`, `palette`. For a PowerPoint file also `background` (the master's background colour, or NULL), `slide_size` ("16:9" or "4:3") and `slide_designs` (`list(title, content)`, each NULL or `list(background, decor, title, subtitle, body)`; a picture of `decor` has `src`, a data URL, and `name`). For a Word file the text sizes it sets (`body_size`, `h1_size`, `h2_size`, `title_size`, `caption_size`), `orientation`, `size` and `margins`.

`quire_to_pdf`

A Word or PowerPoint file as a PDF

Description

A Word or PowerPoint file as a PDF

Usage

```
quire_to_pdf(file, pdf, converter = NULL)
```

Arguments

<code>file</code>	The Word (.docx) or PowerPoint (.pptx) file.
<code>pdf</code>	The PDF to write.
<code>converter</code>	"word", "powerpoint" or "libreoffice"; by default what <code>quire_converter()</code> finds.

Value

`pdf`, invisibly, with attribute `converter`.

`quire_ui`

The report builder in a Shiny app's UI

Description

The report builder in a Shiny app's UI

Usage

```
quire_ui(id, lang = "en", theme = NULL, ...)
```

Arguments

id	The module's id (as given to <code>quire_server()</code>).
lang	The interface's language ("en", "fr", "pt").
theme	The interface's look: a list of design tokens (primary, fontSans, radius, mode = "dark" ...). The report's own design is the report's.
...	Anything else the builder takes (open: a report to open; aiEnabled...).

Value

The UI, to put in the app's page.

quire_word_finish	<i>Finish a Word file in Word</i>
-------------------	-----------------------------------

Description

Microsoft Word opens the file, fills in its contents page and page numbers, embeds its fonts and saves it; with pdf, it also saves the PDF, so the Word file and the PDF are the same document. Windows, with Word installed.

Usage

```
quire_word_finish(docx, pdf = NULL)
```

Arguments

docx	The Word file.
pdf	The PDF to write too, or NULL.

Value

TRUE, invisibly; an error when Word could not.

Index

`htmltools::htmlDependency()`, 3

`quire_can_print`, 2
`quire_chart_options`, 2
`quire_converter`, 3
`quire_converter()`, 12
`quire_dependency`, 3
`quire_dispatch`, 4
`quire_error`, 4
`quire_error()`, 7
`quire_export`, 5
`quire_file_store`, 6
`quire_file_store()`, 7
`quire_host`, 6
`quire_host()`, 4–6, 10
`quire_html_pdf`, 8
`quire_html_pdf()`, 2, 8, 10
`quire_kind`, 8
`quire_kind()`, 7
`quire_plot`, 9
`quire_plot()`, 7
`quire_print_pages`, 10
`quire_print_pages()`, 2, 8
`quire_server`, 10
`quire_server()`, 4, 7, 13
`quire_table`, 11
`quire_table()`, 7
`quire_theme_from_file`, 11
`quire_to_pdf`, 12
`quire_ui`, 12
`quire_ui()`, 3, 10
`quire_word_finish`, 13