

# Package: govukhugo (via r-universe)

January 27, 2025

**Title** GOV.UK style hugo site builder

**Version** 0.0.0.9000

**Description** A set of functions for applying the govuk-hugo theme to hugo static sites and for rendering Rmarkdown documents.

**License** MIT + file LICENSE

**URL** <https://github.com/co-analysis/govuk-hugo-r>

**BugReports** <https://github.com/co-analysis/govuk-hugo-r/issues>

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.0

**Imports** rmarkdown, rstudioapi, htmltools, R.utils, crosstalk, fs, cli, here

**Suggests** blogdown, DT, plotly, leaflet, knitr, ggplot2, svglite, scales

**VignetteBuilder** knitr

**Config/pak/sysreqs** make

**Repository** <https://mattkerlogue.r-universe.dev>

**RemoteUrl** <https://github.com/co-analysis/govukhugo-r>

**RemoteRef** HEAD

**RemoteSha** ae27cf184e2b5470b97388800e13c74a38310d4d

## Contents

build_hugo . . . . .	2
build_hugo_rmd . . . . .	2
clear_filters . . . . .	3
data_dir . . . . .	3
govuk_colours . . . . .	4

govuk_datatable . . . . .	5
init_govuk_hugo . . . . .	6
new_rmd . . . . .	6
render_rmd . . . . .	7
render_svg . . . . .	7
serve_site . . . . .	8
unstrap . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

build_hugo	<i>Build a Hugo site</i>
------------	--------------------------

---

### Description

build\_hugo() Hugo-ifies Rmd files and then calls Hugo to build the static site.

### Usage

```
build_hugo(with_rmd = TRUE, rmd_folder = "R/Rmd", rebuild = FALSE)
```

### Arguments

with_rmd	logical flag for whether to build Rmd files
rmd_folder	path to the folder containing Rmd files
rebuild	whether to rebuild all files

---

build_hugo_rmd	<i>Bulk Hugo-ify Rmd files</i>
----------------	--------------------------------

---

### Description

build\_hugo\_rmd() converts all Rmd files within a folder to Hugo-ified HTML, it will place the Hugo-ified HTML files into the content sections designated in the Rmd YAML front matter. Use [render\\_rmd\(\)](#) if you want to override this. By default it expects Rmd files in the R/Rmd folder. The default behaviour is to only build files that are changed, which is tracked using the rmd.log file that is created/stored in the rmd\_folder, you can force the building of all pages by setting rebuild = TRUE. If working in RStudio the process will save all open source editors windows before running.

### Usage

```
build_hugo_rmd(rmd_folder = "R/Rmd", rebuild = FALSE, save = TRUE)
```

```
govukhugo_knit(input, ...)
```

**Arguments**

rmd_folder	path to the folder containing Rmd files
rebuild	whether to rebuild all Rmd files
save	whether to save open RStudio source files before build
input	recieved Rmd file
...	knitr fluff

**Functions**

- govukhugo\_knit: Don't directly knit govukhugo documents

---

clear_filters	<i>Clear crosstalk filters</i>
---------------	--------------------------------

---

**Description**

Insert a "clear filters" option to a set of filter controls. This adds an HTML div element that works with jQuery in govukhugo to clear crosstalk filters on a page, note that this clears all filters on a page including any DT search boxes.

**Usage**

```
clear_filters(text = "Clear filters")
```

**Arguments**

text	The text to display (defaults to "Clear filters")
------	---

---

data_dir	<i>Get the right data directory</i>
----------	-------------------------------------

---

**Description**

The govukhugo package assumes data is stored in R/data, when processing Rmarkdown files this folder is copied to a temporary folder. This is a helper function to easily work both interactively when editing an Rmarkdown file and when running the build\_hugo\_rmd() function.

**Usage**

```
data_dir(alt_path = NULL)
```

**Arguments**

alt_path	a non-standard location
----------	-------------------------

**Value**

the location of the data folder

---

govuk_colours	<i>Use GOV.UK colours</i>
---------------	---------------------------

---

## Description

Access the GOV.UK design system colour palette

## Usage

```
govuk_colours
```

```
govuk_palette(pal = "categorical")
```

## Arguments

pal	One of <code>categorical</code> (the default), <code>blue</code> , <code>blrd</code> , <code>blrd_dark</code> , <code>blyl</code> , <code>putq</code> , or a colour name from <code>govuk_colours</code>
-----	--

## Format

An object of class `character` of length 34.

## Details

The GOV.UK design system includes a colour palette, these can be accessed via the `govuk_colours` vector, it includes both Sass variable colour names (e.g. `govuk-text-colour`) as well as the named colour palette.

The GOV.UK colour scheme was not designed with data visualisation in mind. The `govuk_palette` function provides an opinionated selection of GOV.UK colours for use in charts (e.g. via [ggplot2::scale\\_fill\\_manual\(\)](#)).

The `categorical` palette (the default) provides an opinionated set of six GOV.UK colours for use in categorical/qualitative palettes.

The `blues` palette orders the GOV.UK colour palettes blues from dark to light and can be used for sequential colour palettes. Alternatively you can supply a name from the `govuk_colours` vector to generate the end-points for a sequential palette using that colour (where the lightest colour is roughly a third lighter than the input colour).

There are also four opinionated palettes for use in a divergent scale, these all use an off-white (the mid-colour between GOV.UK light grey and white)

- `blrd` which use GOV.UK blue and GOV.UK red as its end-points
- `blrd_dark` which uses GOV.UK dark blue as its blue end-point and a darker red of the same hue as GOV.UK red as its red end-point
- `blyl` which uses GOV.UK blue and GOV.UK yellow as its end-points
- `putq` which uses GOV.UK purple and GOV.UK turquoise as its end-points

The categorical and divergent palettes have been checked for general support for users with colour blindness.

**Value**

A set of hexadecimal colours

**Functions**

- govuk\_colours: GOV.UK colour palette

---

govuk_datatable	<i>datatables for GOV.UK</i>
-----------------	------------------------------

---

**Description**

govuk\_datatable is a wrapper for `DT::datatable()` that will enable the govuk-hugo theme to apply GOV.UK CSS classes to the output.

**Usage**

```
govuk_datatable(
  data,
  title = NULL,
  element_id = NULL,
  col_names = NULL,
  page_length = 10,
  search = FALSE,
  small_text = FALSE,
  buttons = TRUE,
  col_defs = NULL,
  copy_info = NULL,
  export_file = NULL,
  options = NULL
)
```

**Arguments**

data	the data frame to display
title	a title/caption for the table
element_id	optionally, an id for the resulting HTML chunk
col_names	optionally, a vector of column names, otherwise the names of the data object
page_length	the number of rows to show per page, default is 10
search	whether to include the search box
small_text	whether to render the table content with a smaller font
buttons	whether to include copy and download buttons
col_defs	optionally, a list of column definitions
copy_info	information to include when copying the file
export_file	the file name for the exported csv
options	optionally, a list of additional options to pass to <code>DT::datatable()</code>

---

init_govuk_hugo	<i>Initiate a Hugo site using the govuk-hugo theme</i>
-----------------	--

---

### Description

init\_govuk\_hugo() calls Hugo to create a new site scaffold, and adds govuk-hugo as a theme. Note that this function only works inside of RStudio with an RStudio project. It is recommended that your project also uses git for version control.

### Usage

```
init_govuk_hugo()
```

---

new_rmd	<i>Open a new editor with the Rmarkdown skeleton</i>
---------	--

---

### Description

The govukhugo package includes a skeleton Rmarkdown file, which can be easily selected via the RStudio GUI. new\_rmd() allows you to open a new RStudio editor with a copy of the skeleton via the console.

### Usage

```
new_rmd(  
  title = NULL,  
  date = NULL,  
  section = NULL,  
  weight = NULL,  
  summary = NULL  
)
```

### Arguments

title	The page title
date	The page date
section	The section the page will go in
weight	The page weight
summary	A summary of the page

---

`render_rmd`*Hugo-ify an R Markdown document*

---

**Description**

Hugo is largely used to convert plain markdown files into HTML, but it also supports HTML documents as input files. `render_rmd()` renders Rmd files using `rmarkdown::html_document()` and then converts the HTML to work with Hugo.

By default `render_rmd()` will read the YAML front-matter of an Rmd for a section parameter and use that as a sub-directory of the content folder as the output location.

**Usage**

```
render_rmd(rmd_file, tmp_dir = tempdir(), out_dir = NULL, quiet = TRUE)
```

**Arguments**

<code>rmd_file</code>	the .rmd file to convert
<code>tmp_dir</code>	a temporary directory for processing
<code>out_dir</code>	override the default output directory
<code>quiet</code>	keep knitr quiet, defaults to TRUE, set to FALSE to see knitr messages

---

`render_svg`*Render ggplot2 objects as in-line SVG*

---

**Description**

`render_svg()` converts ggplot2 objects to SVG code to render them as in-line SVG code, and provides options for providing ARIA labels for assistive technologies.

**Usage**

```
render_svg(  
  plot,  
  width,  
  height,  
  units = "px",  
  alt_title = NULL,  
  alt_desc = NULL,  
  caption = NULL,  
  dpi = 96  
)
```

**Arguments**

plot	A <code>ggplot2::ggplot()</code> object
width	The desired width of the object
height	The desired height of the object
units	The units of width and height, default is px (pixels)
alt_title	Short alt text (will show as tool-tip)
alt_desc	Longer alt text (embedded within SVG)
caption	A caption to display to all users, can also be set to "alt_title" or "alt_desc" to match their value
dpi	Dots per inch, default of 96 for screen resolution, switch to 300 if using physical units (mm, cm or inches)

**Details**

When rendering to HTML, Rmarkdown will render plots as `<img>` tags, including when the output device is set to `svg` or `svglite`. This is problematic for accessibility reasons, as while scalable the plot continues to be rendered by the browser as an image meaning that any text within the plot is not selectable by the user and cannot be read by a screen reader.

`render_svg()` uses `ggplot2::ggsave()` to convert the plot to an chunk of SVG code, rendering text as words (via the `svglite::svglite()` device) and returning that code rather than producing a plot. If the `alt_title`, `alt_desc` are used then these will embedded within the SVG code, these are encoded as ARIA labels for the SVG chunk and `alt_title` will also present as a tooltip. Optionally, a caption can be provided which will be inserted below the plot, if the `caption` argument is set as `alt_title` or `alt_desc` then it will take the value of that argument.

---

serve\_site

*Serve the site*

---

**Description**

These are wrappers around `blogdown::serve_site()` and `blogdown::stop_server()`. The main purpose is to ensure that the blogdown re-knitting process doesn't run when files are saved. `stop_server()` is a simple re-export.

**Usage**

```
serve_site(
  build_rmd = TRUE,
  rmd_folder = "R/Rmd",
  rebuild = FALSE,
  drafts = TRUE,
  future = TRUE
)

stop_server()
```



**Arguments**

build_rmd	whether to build Rmd files before serving
rmd_folder	the folder containing Rmd files
rebuild	whether to trigger a rebuild of all Rmd files
drafts	whether to include drafts
future	whether to include pages with a date in the future

**Functions**

- stop\_server: Stop serving a Hugo site

---

unstrap	<i>Remove bootstrap css from crosstalk components</i>
---------	---

---

**Description**

The crosstalk injects **Bootstrap** into rendered HTML. `unstrap()` removes the Bootstrap dependencies from the components.

**Usage**

```
unstrap(x)
```

**Arguments**

x a crosstalk component, such as `crosstalk::filter_select()`

**Value**

The component x but with any "bootstrap" dependency removed from its `html_dependencies` attribute.

**Examples**

```
# The fs object will inject css into your page.
if (requireNamespace("crosstalk", quietly = TRUE)) {
  df <- crosstalk::SharedData$new(mtcars)

  fs <- crosstalk::filter_select(
    id = "myselector",
    label = "select something",
    sharedData = df,
    group = ~cyl
  )

  # The fs_nobootstrap object won't inject css into your page.
  fs_nobootstrap <- unstrap(fs)
}
```

# Index

## \* datasets

- govuk\_colours, 4
  
- blogdown::serve\_site(), 8
- blogdown::stop\_server(), 8
- build\_hugo, 2
- build\_hugo\_rmd, 2
  
- clear\_filters, 3
- crosstalk::filter\_select(), 9
  
- data\_dir, 3
- DT::datatable(), 5
  
- ggplot2::ggplot(), 8
- ggplot2::ggsave(), 8
- ggplot2::scale\_fill\_manual(), 4
- govuk\_colours, 4
- govuk\_datatable, 5
- govuk\_palette (govuk\_colours), 4
- govukhugo\_knit (build\_hugo\_rmd), 2
  
- init\_govuk\_hugo, 6
  
- new\_rmd, 6
  
- render\_rmd, 7
- render\_rmd(), 2
- render\_svg, 7
- rmarkdown::html\_document(), 7
  
- serve\_site, 8
- stop\_server (serve\_site), 8
- svglite::svglite(), 8
  
- unstrap, 9