

# Package: govdown (via r-universe)

June 5, 2024

**Title** GOV.UK Style Templates for R Markdown

**Version** 0.10.1

**Description** A suite of custom R Markdown formats and templates for authoring web pages styled with the GOV.UK Design System.

**License** MIT + file LICENSE

**URL** <https://ukgovdatascience.github.io/govdown/>

**BugReports** <https://github.com/ukgovdatascience/govdown/issues>

**Imports** rmarkdown

**Suggests** reticulate, testthat (>= 2.1.0), crosstalk

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**SystemRequirements** pandoc (>= 2.0) - <http://pandoc.org>

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

**RemoteUrl** <https://github.com/ukgovdatascience/govdown>

**RemoteRef** HEAD

**RemoteSha** e7f33515df1dcf2d25ee3d2e4bb51c48a3525e12

## Contents

govdown_document . . . . .	2
unstrap . . . . .	4
<b>Index</b>	<b>6</b>

---

govdown\_document      *GOV.UK style HTML template*

---

## Description

A template for rendering R Markdown documents as HTML using the GOV.UK Design System. Can be used for single documents or for websites. font: "sans-serif" favicon: "custom" logo: "images/govdown-logo-white-on-transparent.svg" logo\_url: "index.html" logo\_text: "govdown" page\_title: "page\_title" title: "title" phase: alpha feedback\_url: "https://github.com/ukgovdatascience/govdown/issues"

## Usage

```
govdown_document(
  keep_md = FALSE,
  font = c("sans-serif", "new-transport"),
  favicon = c("none", "custom", "govuk"),
  logo = FALSE,
  logo_url = "",
  logo_text = "Logo text",
  page_title = "Page title",
  title = "Title",
  phase = c("none", "alpha", "beta"),
  feedback_url = "404.html",
  google_analytics = NULL,
  ...
)
```

## Arguments

keep_md	logical, whether to keep the intermediate .md file after rendering.
font	one of "sans-serif" (default) or "new-transport". New Transport must be used when the document or website is published on the GOV.UK domain, otherwise it must not be used.
favicon	one of "govuk" (default) or "custom". For websites only, not standalone documents. If "custom" then image files in the directory "favicon/" will be used. The filenames are: <ul style="list-style-type: none"> <li>• apple-touch-icon-152x152.png</li> <li>• apple-touch-icon-167x167.png</li> <li>• apple-touch-icon-180x180.png</li> <li>• apple-touch-icon.png</li> <li>• favicon.ico</li> <li>• mask-icon.svg</li> </ul> See <code>system.file("rmarkdown/resources/favicon-custom.html", package = "govdown")</code> for how the icons are used. See <code>system.file("rmarkdown/resources/assets/images/" package = "govdown")</code> for the default images. All the files are required.

logo	FALSE (default) or a path to an image file.
logo_url	URL to follow when the logo is clicked.
logo_text	Text to place beside the logo.
page_title	Text to go inside the <title> tags, used in browser toolbars, when making bookmarks and in search engine results.
title	Text to appear in the main part of the bar at the top of every page.
phase	one of "none" (default), "alpha" or "beta" to put an alpha or beta banner indicating the maturity of the service (if it is a service).
feedback_url	URL for feedback, given in the phase banner when phase is "alpha" or "beta".
google_analytics	Google Analytics ID for monitoring traffic to the website.
...	passed on to <code>rmarkdown::html_document_base()</code> , used by Shiny and <code>rmarkdown::run()</code> .

## Details

To configure a standalone document, use the yaml at the top of the .Rmd file.

```

---
output:
  govdown::govdown_document:
    font: "sans-serif"
    favicon: "custom"
    logo: "images/logo.svg"
    logo_url: "https://ukgovdatascience.github.io/govdown/"
    logo_text: "logo_text"
    page_title: "page_title"
    title: "title"
    phase: alpha
    feedback_url: "https://github.com/ukgovdatascience/govdown/issues"
    google_analytics: "UA-12345678-90"
---
```

To configure a website, use a `_site.yml` file instead.

```

output_dir: docs # to host on GitHub pages
navbar:
  - text: "Home"
    href: index.html
  - text: "Tech docs"
    href: tech-docs.html
  - text: "News"
    href: NEWS.html
output:
  govdown::govdown_document:
    font: "sans-serif"
    favicon: "custom"
```

```

logo: "images/govdown-logo-white-on-transparent.svg"
logo_url: "index.html"
logo_text: "govdown"
page_title: "page_title"
title: "title"
phase: alpha
feedback_url: "https://github.com/ukgovdatascience/govdown/issues"
google_analytics: "UA-45097885-11"

```

### Value

R Markdown output format to pass to `rmarkdown::render()`

### Examples

```

## Not run:
# Requires pandoc version 2+
input_rmd <- system.file("extdata/input.Rmd", package = "govdown")
x <- rmarkdown::render(input_rmd, govdown_document())
y <- rmarkdown::render(input_rmd, govdown_document(phase = "alpha"))
utils::browseURL(x)
utils::browseURL(y)

## End(Not run)

```

---

unstrap

*Remove bootstrap css from crosstalk components*

---

### Description

If your page renders weirdly, it might be because a crosstalk component like `crosstalk::filter_select()` is injecting unexpected css into your page, which happens because they try to use the Bootstrap css library by default. The `unstrap()` function prevents that from happening.

### 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

`crosstalk::filter_select()`, 4  
`govdown_document`, 2  
`rmarkdown::html_document_base()`, 3  
`rmarkdown::render()`, 4  
`rmarkdown::run()`, 3  
`unstrap`, 4