Package 'metathis'

May 8, 2026

Title HTML Metadata Tags for 'R Markdown' and 'Shiny'

Version 1.1.4

Description Create meta tags for 'R Markdown' HTML documents and 'Shiny' apps for customized social media cards, for accessibility, and quality search engine indexing. 'metathis' currently supports HTML documents created with 'rmarkdown', 'shiny', 'xaringan', 'pagedown', 'bookdown', and 'flexdashboard'.

License MIT + file LICENSE

URL https://pkg.garrickadenbuie.com/metathis/, https://github.com/gadenbuie/metathis

BugReports https://github.com/gadenbuie/metathis/issues

Imports htmltools, knitr, magrittr, purrr

Suggests rmarkdown, shiny, testthat (>= 2.1.0)

Config/Needs/website gadenbuie/grkgdown

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Garrick Aden-Buie [aut, cre] (ORCID: <https://orcid.org/0000-0002-7111-0077>)

Maintainer Garrick Aden-Buie <garrick@adenbuie.com>

Repository CRAN

Date/Publication 2023-07-11 12:00:09 UTC

Contents

`include_meta` *Include Metadata Tags in HTML Document*

Description

Use `include_meta()` to explicitly declare the `meta()` tags as an HTML dependency. In general, this is not required when knitting to an HTML document. This function explicitly attaches an `htmltools::htmlDependency()` and may work in some unusual cases.

Usage

```
include_meta(.meta)
```

Arguments

`.meta`	A `meta` object created by `meta()` or `as_meta()`, or returned by a `meta_*()` object.

Value

An `htmltools::htmlDependency()` containing the metadata tags to be included in the <head> of the HTML document.

See Also

Other meta_actions: `write_meta()`

Examples

```
meta() %>%
  meta_name("github-repo" = "gadenbuie/metathis") %>%
  include_meta()
```

`meta` *Initialize a List of HTML Metadata Tags*

Description

Initialize a *metathis* object (i.e. a list of HTML metadata tags), test if an object is a *metathis* object, or coerce a list of `meta` tags to be a *metathis* object.

Usage

```
meta()

is_meta(x)

as_meta(x)
```

Arguments

`x`	A list or metathis object

Value

A `meta` object, or a set of <meta> HTML tags inside an HTML <head> tag. For use in `rmarkdown::html_document()`, `shiny::runApp()`, or other HTML locations.

Functions

- `is_meta()`: Test if an objects is a *metathis* object
- `as_meta()`: Convert a list of meta tags into a *metathis* object.

See Also

Other meta: `meta_apple_itunes_app()`, `meta_apple_web_app()`, `meta_general()`, `meta_geo()`, `meta_google_scholar()`, `meta_name()`, `meta_social()`, `meta_tag()`, `meta_viewport()`

Examples

```
meta() %>%
  meta_viewport() %>%
  is_meta()

list_of_meta_tags <- list(
  htmltools::tags$meta(github = "gadenbuie"),
  htmltools::tags$meta(twitter = "grrrck")
)

as_meta(list_of_meta_tags)
```

meta_apple_itunes_app *Apple Smart Banner Meta Tag*

Description

Apple Smart Banner Meta Tag

Usage

```
meta_apple_itunes_app(.meta = meta(), app_id = NULL, affiliate_id = NULL, ...)
```

Arguments

.meta	A meta object created by meta() or as_meta(), or returned by a meta_*() object.
app_id	Apple app ID
affiliate_id	Apple affiliate ID
...	Additional name=value pairs.

Value

A meta object, or a set of <meta> HTML tags inside an HTML <head> tag. For use in rmarkdown::html_document(), shiny::runApp(), or other HTML locations.

Example

```
# <!-- Smart App Banner -->
<meta name="apple-itunes-app" content="app-id=APP_ID,affiliate-data=AFFILIATE_ID,app-argument=SOME_T
```

See Also

Other meta: meta_apple_web_app(), meta_general(), meta_geo(), meta_google_scholar(), meta_name(), meta_social(), meta_tag(), meta_viewport(), meta()

meta_apple_web_app *Apple Web App Meta Tags*

Description

Apple Web App Meta Tags

Usage

```
meta_apple_web_app(
  .meta = meta(),
  title = NULL,
  capable = NULL,
  status_bar_style = c("default", "black", "black-translucent")
)
```

Arguments

.meta	A meta object created by meta() or as_meta(), or returned by a meta_*() object.
title	Launch Icon Title
capable	Enables standalone (full-screen) mode if TRUE
status_bar_style	Status bar appearance. has no effect unless standalone more is enabled (see capable). "If content is set to default, the status bar appears normal. If set to black, the status bar has a black background. If set to black-translucent, the status bar is black and translucent. If set to default or black, the web content is displayed below the status bar. If set to black-translucent, the web content is displayed on the entire screen, partially obscured by the status bar. The default value is default."

Value

A meta object, or a set of <meta> HTML tags inside an HTML <head> tag. For use in rmarkdown::html_document(), shiny::runApp(), or other HTML locations.

References

https://developer.apple.com/library/archive/documentation/AppleApplications/Reference/SafariHTMLRef/Articles/MetaTags.html

See Also

Other meta: meta_apple_itunes_app(), meta_general(), meta_geo(), meta_google_scholar(), meta_name(), meta_social(), meta_tag(), meta_viewport(), meta()

Examples

```
meta() %>%
  meta_apple_web_app(
    title = "My Fancy App",
    capable = TRUE,
    status_bar_style = "black-translucent"
  )
```

meta_general *General Metadata Tags*

Description

Generate metadata tags for general website properties.

Usage

```
meta_general(
  .meta = meta(),
  application_name = NULL,
  theme_color = NULL,
  description = NULL,
  robots = NULL,
  generator = NULL,
  subject = NULL,
  rating = NULL,
  referrer = NULL
)

meta_description(.meta = meta(), description)

meta_subject(.meta = meta(), subject)

meta_referrer(.meta = meta(), referrer)

meta_robots(.meta = meta(), robots)

meta_theme_color(.meta = meta(), theme_color)
```

Arguments

.meta	A meta object created by meta() or as_meta(), or returned by a meta_*() object.
application_name	Name of web application (only should be used if the website is used as an app).
theme_color	Theme Color for Chrome, Firefox OS and Opera, e.g. "#00589a".
description	Short description of the document (limit to 150 characters), This content *may* be used as a part of search engine results.
robots	Control the behavior of search engine crawling and indexing, e.g. "index,follow". Valid names are "index", "noindex", "follow", "nofollow". May be a vector or a single string with comma-separated values. See https://www.robotstxt.org/meta.html for more information.
generator	Identify the software used to build the document (i.e. - WordPress, Dreamweaver).

subject	Short description of your document's subject.
rating	Gives a general age rating based on the document's content, e.g. "General".
referrer	Allows control over how referrer information is passed, .e.g. "no-referrer". Valid values include "no-referrer", "no-referrer-when-downgrade", "same-origin", "origin", "strict-origin", "origin-when-cross-origin", "strict-origin-when-cross-origin" or "unsafe-url".

Value

A meta object, or a set of <meta> HTML tags inside an HTML <head> tag. For use in rmarkdown::html_document(), shiny::runApp(), or other HTML locations.

See Also

Other meta: meta_apple_itunes_app(), meta_apple_web_app(), meta_geo(), meta_google_scholar(), meta_name(), meta_social(), meta_tag(), meta_viewport(), meta()

Examples

```
meta() %>%
  meta_general(
    application_name = "Application Name",
    theme_color = "#4285f4",
    description = "A description of this page",
    robots = "index,follow",
    generator = "R-Shiny",
    subject = "Awesome R projects",
    rating = "General",
    referrer = "no-referrer"
  )
```

---

meta_geo *Geotagging Metadata Tags*

Description

Geotagging Metadata Tags

Usage

```
meta_geo(
  .meta = meta(),
  icbm = NULL,
  geo_position = NULL,
  geo_region = NULL,
  geo_placename = NULL,
  ...
)
```

Arguments

.meta	A meta object created by meta() or as_meta(), or returned by a meta_*() object.
icbm	Latitude and longitude of geographic positions specified as "lat, long". Can optionally be a length-two vector, i.e. c(lat, long).
geo_position	Latitude and longitude of geographic positions specified as "lat;long". Can optionally be a length-two vector, i.e. c(lat, long).
geo_region	Name of the geographic region related to the page content, specified using ISO-3166 2-character country code and 2-character national subdivision. Example: "US-NY".
geo_placename	Name of the geographic place related to the page content. Example: "Atlanta, Georgia".
...	Additional geotagging metadata keyword and value pairs, such as geo.country, geo.a1, ..., geo.a3, etc. Underscores in the keyword will be converted to periods, so you can also specify geo_country in place of geo.country.

Value

A meta object, or a set of <meta> HTML tags inside an HTML <head> tag. For use in rmarkdown::html_document(), shiny::runApp(), or other HTML locations.

References

ICBM on Wikipedia, Geotagging on Wikipedia

See Also

Other meta: meta_apple_itunes_app(), meta_apple_web_app(), meta_general(), meta_google_scholar(), meta_name(), meta_social(), meta_tag(), meta_viewport(), meta()

Examples

```
meta() %>%
  meta_geo(
    icbm = c(50.167958, -97.133185),
    geo_position = c(50.167958, -97.133185),
    geo_placename = "Manitoba, Canada",
    geo_region = "ca-mb"
  )
```

`meta_google_scholar` *Add Google Scholar Metadata*

Description

Add bibliographic metadata to pages in the format expected by Google Scholar. Please reference the Google Scholar Inclusion page for the most up-to-date information and instructions. Note that this function adds the `citation_` prefix to all of its arguments; the `title` argument becomes the `citation_title` `<meta>` tag.

Usage

```
meta_google_scholar(
  .meta = meta(),
  title,
  author,
  publication_date,
  online_date = NULL,
  journal_title = NULL,
  conference_title = NULL,
  volume = NULL,
  issue = NULL,
  firstpage = NULL,
  lastpage = NULL,
  pdf_url = NULL,
  issn = NULL,
  isbn = NULL,
  dissertation_institution = NULL,
  technical_report_institution = NULL,
  technical_report_number = NULL
)
```

Arguments

`.meta`	A `meta` object created by `meta()` or `as_meta()`, or returned by a `meta_*()` object.
`title`	The title of the paper. The title tag must contain the title of the paper. Don't use it for the title of the journal or a book in which the paper was published, or for the name of your repository. This tag is required for inclusion in Google Scholar.
`author`	A vector of author names. The `author` tag, must contain the authors (and only the actual authors) of the paper. Don't use it for the author of the website or for contributors other than authors, e.g., thesis advisors. Author names can be listed either as "Smith, John" or as "John Smith". Put each author name in a separate tag and omit all affiliations, degrees, certifications, etc., from this field. At least one author tag is required for inclusion in Google Scholar.

`publication_date`, `online_date`

The date the paper was published in the journal (`publication_date`) or published online (`online_date`).

The `publication_date` tag must contain the date of publication, i.e., the date that would normally be cited in references to this paper from other papers. Don't use it for the date of entry into the repository - that should go into `online_date` instead. Provide full dates in the "2010/5/12" format if available; or a year alone otherwise. This tag is required for inclusion in Google Scholar.

`journal_title`, `conference_title`, `issn`, `isbn`, `volume`, `issue`, `firstpage`, `lastpage`

For journal and conference papers, provide the remaining bibliographic citation data in the following tags: `journal_title` or `conference_title`, `issn`, `isbn`, `volume`, `issue`, `firstpage`, and `lastpage`. These fields must contain sufficient information to identify a reference to this paper from another document, which is normally all of: (a) journal or conference name, (b) volume and issue numbers, if applicable, and (c) the number of the first page of the paper in the volume (or issue) in question.

`pdf_url`

The `<meta>` tags normally apply only to the exact page on which they're provided. If this page shows only the abstract of the paper and you have the full text in a separate file, e.g., in the PDF format, please specify the locations of all full text versions using `pdf_url`. The content of the tag is the absolute URL of the PDF file; for security reasons, it must refer to a file in the same subdirectory as the HTML abstract.

`dissertation_institution`, `technical_report_institution`, `technical_report_number`

For theses, dissertations, and technical reports, provide the remaining bibliographic citation data in the following tags: `dissertation_institution`, `technical_report_institut` for the name of the institution and `technical_report_number` for the number of the technical report. As with journal and conference papers, you need to provide sufficient information to recognize a formal citation to this document from another article.

Value

A `meta` object, or a set of `<meta>` HTML tags inside an HTML `<head>` tag. For use in `rmarkdown::html_document()`, `shiny::runApp()`, or other HTML locations.

References

https://scholar.google.com/intl/en/scholar/inclusion.html#indexing

See Also

Other meta: `meta_apple_itunes_app()`, `meta_apple_web_app()`, `meta_general()`, `meta_geo()`, `meta_name()`, `meta_social()`, `meta_tag()`, `meta_viewport()`, `meta()`

Examples

```
meta_google_scholar(
```

```
  title = c(
    "The testis isoform of the phosphorylase kinase catalytic subunit (PhK-T)",
    "plays a critical role in regulation of glycogen mobilization in developing lung"
  ),
  author = c(
    "Liu, Li",
    "Rannels, Stephen R.",
    "Falconieri, Mary",
    "Phillips, Karen S.",
    "Wolpert, Ellen B.",
    "Weaver, Timothy E."
  ),
  publication_date = "1996/05/17",
  journal_title = "Journal of Biological Chemistry",
  volume = 271,
  issue = 20,
  firstpage = 11761,
  lastpage = 11766,
  pdf_url = "http://www.example.com/content/271/20/11761.full.pdf"
)
```

`meta_name` *Create name/content metadata tag pairs*

Description

Creates metadata tag pairs where the arguments are the name values and their values are content values.

Usage

```
meta_name(.meta = meta(), ...)
```

Arguments

`.meta`	A `meta` object created by `meta()` or `as_meta()`, or returned by a `meta_*()` object.
`...`	Name (argument names) and content (argument value) pairs.

Value

A `meta` object, or a set of <meta> HTML tags inside an HTML <head> tag. For use in `rmarkdown::html_document()`, `shiny::runApp()`, or other HTML locations.

See Also

Other meta: `meta_apple_itunes_app()`, `meta_apple_web_app()`, `meta_general()`, `meta_geo()`, `meta_google_scholar()`, `meta_social()`, `meta_tag()`, `meta_viewport()`, `meta()`

Examples

```
meta() %>%
  meta_name("github-repo" = "hadley/r4ds")
```

`meta_social` *Social Media Metadata*

Description

Generate metadata tags for social media cards.

Usage

```
meta_social(
  .meta = meta(),
  title = NULL,
  url = NULL,
  image = NULL,
  image_alt = NULL,
  image_width = NULL,
  image_height = NULL,
  description = NULL,
  twitter_card_type = c("summary", "summary_large_image", "app", "player"),
  twitter_creator = NULL,
  twitter_site = twitter_creator,
  og_type = "website",
  og_locale = "en_US",
  og_author = NULL,
  og_site_name = NULL,
  facebook_app_id = NULL,
  disable_pinterest = FALSE
)
```

Arguments

`.meta`	A `meta` object created by `meta()` or `as_meta()`, or returned by a `meta_*()` object.
`title`	Content title
`url`	Content URL
`image`	Image url for card.
`image_alt`	A description of what's in the image (not a caption)
`image_width`	The width of the image in pixels
`image_height`	The height of the image in pixels

description — Content description. If you have already used meta_general() to set the content's description, that description will automatically be used.

twitter_card_type — One of "summary", "summary_large_image", "app", or "player".

twitter_creator — @username for the content creator / author.

twitter_site — @username`` for the website used in the card footer. twitter_creator‘ is used by default.

og_type — Open Graph card type, default is "website". Other common options include "article", "book", or "profile". The full list of valid options can be referenced at https://ogp.me/

og_locale — The locale these tags are marked up in. Of the format language_TERRITORY. Default is "en_US".

og_author — Writers of the article. Multiple authors may be specified in a vector of character strings.

og_site_name — The name of the site hosting the content

facebook_app_id — The Facebook app ID. See the Facebook Open Graph Markup page for more information.

disable_pinterest — If TRUE, adds a metadata tag disabling pins from your website. See the Pinterest help center for more information.

Value

A meta object, or a set of <meta> HTML tags inside an HTML <head> tag. For use in rmarkdown::html_document(), shiny::runApp(), or other HTML locations.

References

- Open Graph
- Google Structured Data Testing Tool
- Facebook Sharing Debugger
- Twitter Card Documentation

See Also

Other meta: meta_apple_itunes_app(), meta_apple_web_app(), meta_general(), meta_geo(), meta_google_scholar(), meta_name(), meta_tag(), meta_viewport(), meta()

Examples

```
meta() %>%
  meta_social(
    title = "R for Data Science",
    description = "This book with teach you how to do data science with R",
    url = "https://r4ds.had.co.nz",
```

```
  image = "https://r4ds.had.co.nz/cover.png",
  image_alt = "The cover of the R4DS book",
  og_type = "book",
  og_author = c("Garrett Grolemund", "Hadley Wickham"),
  twitter_card_type = "summary",
  twitter_creator = "@hadley"
)
```

`meta_tag` *Create a metadata tag for attribute/value pairs*

Description

Creates a <meta> tag for attribute value pairs, where argument names correspond to attribute names.

Usage

```
meta_tag(.meta = meta(), ...)
```

Arguments

`.meta`	A `meta` object created by `meta()` or `as_meta()`, or returned by a `meta_*()` object.
`...`	Attribute names and values as `attribute = value`. Values must be a single character string.

Value

A `meta` object, or a set of <meta> HTML tags inside an HTML <head> tag. For use in `rmarkdown::html_document()`, `shiny::runApp()`, or other HTML locations.

See Also

Other meta: `meta_apple_itunes_app()`, `meta_apple_web_app()`, `meta_general()`, `meta_geo()`, `meta_google_scholar()`, `meta_name()`, `meta_social()`, `meta_viewport()`, `meta()`

Examples

```
meta() %>%
  meta_tag(
    "http-equiv" = "Content-Security-Policy",
    content = "default-src 'self'"
  )
```

meta_viewport *Viewport Meta Tag*

Description

Create or add a viewport meta tag.

Usage

```
meta_viewport(
  .meta = meta(),
  width = "device-width",
  initial_scale = "1",
  orientation = c("auto", "portrait", "landscape"),
  min_width = NULL,
  max_width = NULL,
  height = NULL,
  min_height = NULL,
  max_height = NULL,
  minimum_scale = NULL,
  maximum_scale = NULL,
  user_scalable = NULL,
  ...
)
```

Arguments

.meta	A meta object created by meta() or as_meta(), or returned by a meta_*() object.
width	Sets the width of initial viewport. width sets min_width and max_width and may contain two values; for example, the following are equivalent: "300px 500px" or c("300px", "500px")). The values may be a number with units, a percentage, or "device-width". Pixels are assumed if no units are provided.
initial_scale	Initial scale
orientation	One of "auto", "portrait", "landscape".
min_width, max_width	Minimum and maximum initial viewport width. See width for more information. width is ignored if min_width or max_width are set.
height, min_height, max_height	Sets height of initial viewport. Follows the same conventions as width, min_width, and max_width.
minimum_scale	Minimum scale
maximum_scale	Maximum scale
user_scalable	User scalable
...	Additional name/value pairs

Value

A meta object, or a set of <meta> HTML tags inside an HTML <head> tag. For use in rmarkdown::html_document(), shiny::runApp(), or other HTML locations.

References

MDN: Viewport Meta Tag,

See Also

Other meta: meta_apple_itunes_app(), meta_apple_web_app(), meta_general(), meta_geo(), meta_google_scholar(), meta_name(), meta_social(), meta_tag(), meta()

Examples

```
meta() %>%
  meta_viewport()

meta() %>%
  meta_viewport(orientation = NULL)

meta() %>%
  meta_viewport(maximum_scale = 1)
```

---

write_meta *Write Metadata Tags to a File*

Description

Write your metadata tags to an HTML file that can be manually included in your page.

Usage

```
write_meta(.meta, path = "meta.html", append = FALSE)
```

Arguments

.meta	A meta object created by meta() or as_meta(), or returned by a meta_*() object.
path	The file to write into, defaults to "meta.html".
append	logical. Only used if the argument file is the name of file (and not a connection or "|cmd"). If TRUE output will be appended to file; otherwise, it will overwrite the contents of file.

Value

Returns .meta invisibly.

See Also

Other meta_actions: include_meta()

Examples

```
meta_html_snippet <- tempfile("metathis_example", fileext = ".html")

meta() %>%
  meta_name("package" = "metathis") %>%
  write_meta(meta_html_snippet)

readLines(meta_html_snippet, warn = FALSE)
```

Index