Package 'plume'

September 27, 2025

Type Package

Title A Simple Author Handler for Scientific Writing

Version 0.3.0

Description Handles and formats author information in scientific writing
in 'R Markdown' and 'Quarto'. 'plume' provides easy-to-use and
flexible tools for inserting author data in 'YAML' as well as
generating author and contribution lists (among others) as strings
from tabular data.

License GPL (>= 3)

URL <https://arnaudgallou.github.io/plume/>,
<https://github.com/arnaudgallou/plume>

BugReports <https://github.com/arnaudgallou/plume/issues>

Depends R (>= 4.1.0)

Imports dplyr (>= 1.0.0), glue (>= 1.3.2), jsonlite (>= 1.6.0), knitr
(>= 1.40), lifecycle (>= 1.0.3), purrr (>= 1.0.0), R6, readr
(>= 1.0.0), rlang (>= 1.0.0), stringr, tibble (>= 3.0.0), tidyr
(>= 1.1.0), tidyselect (>= 1.0.0), vctrs (>= 0.3.0), yaml (>=
2.3.8)

Suggests covr, fontawesome, gt, rmarkdown, testthat, waldo (>= 0.3.0),
withr

VignetteBuilder knitr

Config/Needs/website arnaudgallou/cygne

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Arnaud Gallou [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-1002-4247>>)

Contents

as_lines	*Print vector elements on distinct lines*

Description

Thin wrapper around `cat()` to display vector elements on distinct lines when rendering an R Markdown or Quarto document.

Usage

```
as_lines(...)
```

Arguments

`...`   Objects to print.

Value

`NULL`, invisibly.

Examples

```
aut <- Plume$new(encyclopedists)
as_lines(aut$get_affiliations())
```

---

credit_roles	*CRediT roles*

Description

Helper function returning the 14 contributor roles of the Contributor Roles Taxonomy (CRediT). This function is the default argument of the roles and role_cols parameters in plume classes and plm_template(), respectively.

Usage

```
credit_roles(oxford_spelling = TRUE)
```

Arguments

oxford_spelling

      Should the suffix -ize/-ization be used?

Value

A named vector.

Examples

```
credit_roles()
```

---

encyclopedists	*Famous encyclopedists*

Description

Data on four notable authors of the Encyclopédie (originally "Encyclopédie, ou dictionnaire raisonné des sciences, des arts et des métiers") published in France in the second half of the 18th century. The dataset is available in English (encyclopedists) and French (encyclopedists_fr).

Usage

```
encyclopedists
```

```
encyclopedists_fr
```

Format

A tibble with 4 rows and 10 variables:

given_name,prénom authors' given names

family_name,nom authors' family names

email,courriel authors' email addresses

phone,téléphone authors' phone numbers

orcid authors' ORCID

affiliation_1,affiliation_2 authors' affiliations

supervision authors that supervised the project

writing,rédaction authors involved in the writing

note special notes about authors

Examples

```
encyclopedists

encyclopedists_fr
```

---

enumerate   *Enumerate vector elements*

Description

Wrapper around [glue_collapse()](#) using sep = ", " and last = " and " as default arguments.

Usage

```
enumerate(x, sep = ", ", last = " and ")
```

Arguments

x	A character vector.
sep	Separator used to separate the terms.
last	Separator used to separate the last two items if x has at least 2 items.

Value

A character string with the same class as x.

Examples

```
aut <- Plume$new(encyclopedists)
enumerate(aut$get_author_list())
```

---

icn_orcid	*Create an ORCID icon*

Description

Create an ORCID icon for use with the [`Plume`](Plume) class. This is only supported in R Markdown.

Usage

```
icn_orcid(size = 16, bw = FALSE)
```

Arguments

size	Size of the icon (in pixels).
bw	Should the black and white version of the icon be used?

Value

A plume icon, i.e. an object with S3 class `plm_icon`.

Examples

```
aut <- Plume$new(encyclopedists, orcid_icon = icn_orcid(bw = TRUE))
```

---

plm_symbols	*Set symbols for* Plume

Description

Set the symbols used in a [`Plume`](Plume) object.

Usage

```
plm_symbols(
  affiliation = NULL,
  corresponding = "*",
  note = c("†", "‡", "§", "¶", "#", "**")
)
```

Arguments

affiliation, corresponding, note

>   Character vectors of symbols to use, or `NULL` to use numerals.

Value

A named list.

Examples

```
aut <- Plume$new(
  encyclopedists,
  symbols = plm_symbols(affiliation = letters)
)
aut$get_author_list("^a^")
```

---

plm_template	_Create a table template for plume classes_

Description

Create an empty [tibble](#) that can be used as a template to supply author data.

Usage

```
plm_template(minimal = TRUE, role_cols = credit_roles(), credit_roles = FALSE)
```

Arguments

`minimal`	If `TRUE`, returns an empty tibble with the following columns: `given_name`, `family_name`, `email`, `orcid`, `affiliation` and `note`. Otherwise the function returns a template with all columns that can be supplied to plume classes that are not `PlumeQuarto`-specific.
`role_cols`	A vector of names defining role columns to create. If the vector contains key-value pairs, columns will be named after the keys.
`credit_roles`	**[Deprecated]** It is now recommended to use `role_cols = credit_roles()` to use the [Contributor Roles Taxonomy](#).

Value

An empty tibble.

Examples

```
plm_template()

plm_template(role_cols = paste0("role_", 1:5))
```

Plume   *Generate author information within a document*

Description

`Plume` provides several methods to generate author information directly within an R Markdown or Quarto document. This is a convenient solution when you don't need preformatted reports.

In some cases, `Plume` gives you greater control over the formatting of author information, as it supports features not available in `PlumeQuarto`.

Notes

`new_plume()` is an alias for `Plume$new()`.

Super classes

`plume::NameHandler` -> `plume::PlumeHandler` -> `plume::StatusSetter` -> `plume::StatusSetterPlume` -> `Plume`

Methods

Public methods:

- `Plume$new()`
- `Plume$get_author_list()`
- `Plume$get_affiliations()`
- `Plume$get_notes()`
- `Plume$get_orcids()`
- `Plume$get_contact_details()`
- `Plume$get_contributions()`
- `Plume$clone()`

Method `new()`: Create a `Plume` object.

Usage:
```
Plume$new(
  data,
  names = NULL,
  roles = credit_roles(),
  symbols = plm_symbols(),
  credit_roles = FALSE,
  initials_given_name = FALSE,
  dotted_initials = TRUE,
  family_name_first = FALSE,
  distinct_initials = FALSE,
  interword_spacing = TRUE,
  orcid_icon = icn_orcid(),
  by = NULL
)
```

Arguments:

data A data frame containing author data.

names A vector of key-value pairs specifying custom names to use, where keys are default names and values their respective replacements.

roles A vector of key-value pairs defining roles where keys identify role columns and values describe the actual roles to use.

symbols Symbols, as defined by `plm_symbols()`, used to link authors to their metadata. Special Markdown characters are automatically escaped internally.

credit_roles **[Deprecated]**
    It is now recommended to use roles = credit_roles() to use the Contributor Roles Taxonomy.

initials_given_name Should the initials of given names be used?

dotted_initials Should initials be dot-separated?

family_name_first Should literal names show family names first?

distinct_initials If `TRUE`, will expand identical initials with additional letters from the last word of their respective family name until initials are unique. Initials of authors sharing the exact same name will remain in the short form.

interword_spacing Should literal names use spacing? This parameter is only useful for people writing in languages that don't separate words with a space such as Chinese or Japanese.

orcid_icon The ORCID icon, as defined by `icn_orcid()`, to be used. Only supported in R Markdown.

by A character string defining the default variable used to assign specific metadata to authors in all set_*() methods. By default, uses authors' id.

Returns: A `Plume` object.

Method `get_author_list()`: Get the list of authors.

Usage:

`Plume$get_author_list(suffix = NULL, format = deprecated())`

Arguments:

suffix A character string defining the format of symbols suffixing author names. See details.

format **[Deprecated]**
    Please use the parameter `suffix` instead.

Details: `suffix` lets you choose which symbol categories to suffix authors with, using the following keys:

- a for affiliations
- c for corresponding authors
- n for notes
- o for ORCID icons (only supported in R Markdown)

The order of the keys determines the order of symbol types. For example, `"ac"` shows affiliation ids first and corresponding author mark second, when `"ca"` shows corresponding author mark first and affiliation ids second. Use `","` to separate and `"^"` to superscript symbols.
Set to `NULL` or `""` to list authors without suffixes.

Returns: A character vector.

Method `get_affiliations()`: Get authors' affiliations.

Usage:
```
Plume$get_affiliations(superscript = TRUE, sep = "")
```
Arguments:

`superscript` Should affiliation ids be superscripted?

`sep` Separator used to separate affiliation ids and affiliations.

Returns: A character vector.

Method `get_notes()`: Get authors' notes.

Usage:
```
Plume$get_notes(superscript = TRUE, sep = "")
```
Arguments:

`superscript` Should note ids be superscripted?

`sep` Separator used to separate note ids and notes.

Returns: A character vector.

Method `get_orcids()`: Get authors' ORCID.

Usage:
```
Plume$get_orcids(compact = FALSE, icon = TRUE, sep = "")
```
Arguments:

`compact` Should links only display the 16-digit identifier?

`icon` Should the ORCID icon be shown? This is only supported in R Markdown.

`sep` Separator used to separate authors and their respective ORCID.

Returns: A character vector.

Method `get_contact_details()`: Get the contact details of corresponding authors.

Usage:
```
Plume$get_contact_details(
  template = "{details} ({name})",
  email = TRUE,
  phone = FALSE,
  fax = FALSE,
  url = FALSE,
  sep = ", ",
  format = deprecated()
)
```
Arguments:

`template` A glue specification that uses the variables `name` and/or `details`.

`email, phone, fax, url` Arguments equal to `TRUE` are evaluated and passed to the variable `details`. By default, only `email` is set to `TRUE`.

`sep` Separator used to separate `details` items.

`format` **[Deprecated]**
    Please use the parameter `template` instead.

Returns: A character vector.

Method `get_contributions()`: Get authors' contributions.

Usage:
```
Plume$get_contributions(
  roles_first = TRUE,
  by_author = FALSE,
  alphabetical_order = FALSE,
  literal_names = FALSE,
  divider = ": ",
  sep = ", ",
  sep_last = " and ",
  dotted_initials = deprecated()
)
```

Arguments:

`roles_first` If `TRUE`, displays roles first and authors second. If `FALSE`, roles follow authors.

`by_author` Should roles be grouped by author?

`alphabetical_order` Should authors be listed in alphabetical order? By default, lists authors in the order they are defined in the data.

`literal_names` Should literal names be used?

`divider` Separator used to separate roles from authors.

`sep` Separator used to separate roles or authors.

`sep_last` Separator used to separate the last two roles or authors if more than one item is associated to a role or author.

`dotted_initials` **[Deprecated]**
   Please use the `dotted_initials` parameter of `Plume$new()` instead.

Returns: A character vector.

Method `clone()`: The objects of this class are cloneable with this method.

Usage:
```
Plume$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
# Create a Plume instance
aut <- Plume$new(encyclopedists)

# Set the desired corresponding authors, using
# authors' id by default
aut$set_corresponding_authors(1, 4)

# Listing authors, followed by affiliation ids
# and the corresponding author mark:
aut$get_author_list("^a,c^")
```

```
# Or maybe with the corresponding author mark
# coming before affiliation ids:
aut$get_author_list("^c,a^")

# Getting more author metadata
aut$get_affiliations()

aut$get_contributions()

# Use `symbols` to change the default symbols.
# E.g. to use letters as affiliation ids:
aut <- Plume$new(
  encyclopedists,
  symbols = plm_symbols(affiliation = letters)
)

aut$get_author_list("^a^")

aut$get_affiliations()

# It is also possible to output contributions in the
# active voice
aut <- Plume$new(
  encyclopedists,
  roles = c(
    supervision = "supervised the project",
    writing = "contributed to the Encyclopédie"
  )
)
aut$get_contributions(roles_first = FALSE, divider = " ")
```

PlumeQuarto   *Add or update author data in YAML for Quarto*

Description

PlumeQuarto allows you to add or update author data in YAML files or the YAML header of Quarto documents. The generated YAML complies with Quarto's author and affiliations schemas. Use this class when working with journal templates.

Notes

new_plume_quarto() is an alias for PlumeQuarto$new().

Super classes

plume::NameHandler -> plume::PlumeHandler -> plume::StatusSetter -> plume::StatusSetterPlumeQuarto -> PlumeQuarto

Methods

Public methods:

- `PlumeQuarto$new()`
- `PlumeQuarto$to_yaml()`
- `PlumeQuarto$clone()`

Method `new()`: Create a `PlumeQuarto` object.

Usage:

```
PlumeQuarto$new(
  data,
  file,
  names = NULL,
  roles = credit_roles(),
  credit_roles = FALSE,
  initials_given_name = FALSE,
  dotted_initials = TRUE,
  by = NULL
)
```

Arguments:

`data` A data frame containing author data.

`file` A `.qmd`, `.yml` or `.yaml` file to insert author data into.

`names` A vector of key-value pairs specifying custom names to use, where keys are default names and values their respective replacements.

`roles` A vector of key-value pairs defining roles where keys identify columns and values describe the actual roles to use.

`credit_roles` **[Deprecated]**
It is now recommended to use `roles = credit_roles()` to use the Contributor Roles Taxonomy.

`initials_given_name` Should the initials of given names be used?

`dotted_initials` Should initials be dot-separated?

`by` A character string defining the default variable used to assign specific metadata to authors in all `set_*()` methods. By default, uses authors' id.

Returns: A `PlumeQuarto` object.

Method `to_yaml()`: Add or update author data in the input `file`.

Usage:

```
PlumeQuarto$to_yaml()
```

Returns: The input `file`, invisibly.

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
PlumeQuarto$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```r
# Create a simple temporary YAML file containing a title
tmp_file <- tempfile(fileext = ".yml")
readr::write_lines("title: Encyclopédie", tmp_file)

# View the temporary file
cat(readr::read_file(tmp_file))

# Create a PlumeQuarto instance using the temporary file
# we've just created
aut <- PlumeQuarto$new(
  encyclopedists,
  file = tmp_file
)

# And add author data to the YAML file
aut$to_yaml()

cat(readr::read_file(tmp_file))

# Running the method again with new data updates the YAML
# accordingly
aut <- PlumeQuarto$new(
  dplyr::slice(encyclopedists, 2),
  file = tmp_file
)
aut$to_yaml()

cat(readr::read_file(tmp_file))

# Clean up
unlink(tmp_file)
```

---

sequential	*Control the sequencing behaviour of character vectors*

Description

Modifier function used to generate logical sequences of characters.

Usage

```r
sequential(x)
```

Arguments

x   A character vector.

Value

A character vector with parent S3 class `sequential`.

Examples

```
aut <- Plume$new(
  tibble::tibble(
    given_name = "X",
    family_name = "Y",
    affiliation = 1:60
  ),
  symbols = plm_symbols(affiliation = sequential(letters))
)

aut$get_affiliations(sep = ": ", superscript = FALSE)
```

---

set_default_names   *Set new default names to a plume subclass*

Description

This function allows you to set new default names to a plume subclass, for example to set default names to a language other than English.

Usage

```
set_default_names(..., .plume_quarto = FALSE)
```

Arguments

`...`   Key-value pairs where keys are default names and values their respective replacements.

`.plume_quarto`   Are you setting new names for [PlumeQuarto](#)?

Details

See `vignette("plume")` for the list of available names.

Value

A named list.

Examples

```
# Extending `Plume` with default names in French
PlumeFr <- R6::R6Class(
  classname = "PlumeFr",
  inherit = Plume,
  private = list(
    names = set_default_names(
      initials = "initiales",
      literal_name = "nom_complet",
      corresponding = "correspondant",
      given_name = "prénom",
      family_name = "nom",
      email = "courriel",
      phone = "téléphone"
    )
  )
)

PlumeFr$new(encyclopedists_fr)
```

Index