

Package ‘vitae’

November 11, 2020

Title Curriculum Vitae for R Markdown

Version 0.3.0

Description

Provides templates and functions to simplify the production and maintenance of curriculum vitae.

Depends R (>= 3.5.0)

Imports rlang, glue, dplyr, rmarkdown (>= 1.18), knitr, xfun, yaml,
tibble, vctrs, pillar

Suggests covr, rorcid, testthat, stringr, htmltools

License GPL-3

Encoding UTF-8

BugReports <https://github.com/mitchelloharawild/vitae/issues>

URL <https://pkg.mitchelloharawild.com/vitae/>,
<https://github.com/mitchelloharawild/vitae>

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Mitchell O'Hara-Wild [aut, cre]
(<<https://orcid.org/0000-0001-6729-7695>>),
Rob Hyndman [aut] (<<https://orcid.org/0000-0002-2140-5352>>),
Yihui Xie [ctb] (<<https://orcid.org/0000-0003-0645-5666>>),
Albert Krewinkel [cph] (Multiple bibliographies lua filter),
JooYoung Seo [ctb] (<<https://orcid.org/0000-0002-4064-6012>>)

Maintainer Mitchell O'Hara-Wild <mail@mitchelloharawild.com>

Repository CRAN

Date/Publication 2020-11-11 15:40:03 UTC

R topics documented:

awesomercv	2
bibliography_entries	3
brief_entries	4
csl_date	4
csl_name	5
cv_document	5

Index	6
--------------	----------

awesomercv	<i>Output formats for CVs</i>
------------	-------------------------------

Description

Each function is a wrapper for `cv_document` to produce curriculum vitae

Usage

```
awesomercv(..., latex_engine = "xelatex")

hyndman(...)

latexcv(..., theme = c("classic", "modern", "rows", "sidebar", "two_column"))

moderncv(
  ...,
  theme = c("casual", "classic", "oldstyle", "banking", "fancy"),
  latex_engine = "pdflatex"
)

twentyseconds(...)
```

Arguments

...	Arguments passed to <code>cv_document</code> .
latex_engine	LaTeX engine for producing PDF output. Options are "pdflatex", "lualatex", and "xelatex".
theme	The theme used for the template.

Value

An R Markdown output format object.

Author(s)

Mitchell O'Hara-Wild and Rob J Hyndman

bibliography_entries *Print bibliography section*

Description

Given a bibliography file, this function will generate bibliographic entries for one or more types of bib entry.

Usage

```
bibliography_entries(file, startlabel = NULL, endlabel = NULL)
```

Arguments

file	A path to a bibliography file understood by <code>rmarkdown::pandoc_citeproc_convert()</code> .
startlabel	Defunct.
endlabel	Defunct.

Value

A dataset representing the bibliographic entries, suitable for generating a reference section in a document.

Author(s)

Rob J Hyndman & Mitchell O'Hara-Wild

Examples

```
# Create a bibliography from a set of packages
bib <- tempfile(fileext = ".bib")
knitr::write_bib(c("vitae", "tibble"), bib)

# Import the bibliography entries into a CV
bibliography_entries(bib)

# The order of these entries can be customised using `dplyr::arrange()`
bibliography_entries(bib) %>%
  arrange(desc(title))

# For more complex fields like author, you can also sort by component fields.
# For example, use `author$family` to sort by family names.
bibliography_entries(bib) %>%
  arrange(desc(author$family))
```

brief_entries	<i>CV entries</i>
---------------	-------------------

Description

This function accepts a data object (such as a tibble) and formats the output into a suitable format for the template used. The inputs can also involve further calculations, which will be done using the provided data.

Usage

```
brief_entries(data, what, when, with, .protect = TRUE)
```

```
detailed_entries(data, what, when, with, where, why, .protect = TRUE)
```

Arguments

data	A data.frame or tibble.
what	The primary value of the entry (such as workplace title or degree).
when	The time of the entry (such as the period spent in the role).
with	The company or organisation.
.protect	When TRUE, inputs to the previous arguments will be protected from being parsed as LaTeX code.
where	The location of the entry.
why	Any additional information, to be included as dot points. Each entry for why is provided in long form (where the what, when, with, and where is duplicated)

Details

All non-data inputs are optional, and will result in an empty space if omitted.

csl_date	<i>A list of dates conforming to the CSL schema</i>
----------	---

Description

This class provides helper utilities to display, sort, and select attributes from a date in the CSL format.

Usage

```
csl_date(x)
```

Arguments

x A list of dates conforming to the structure outlined in the CSL schema

See Also

<https://citeproc-js.readthedocs.io/en/latest/csl-json/markup.html#date-fields>

csl_name *A list of names conforming to the CSL schema*

Description

This class provides helper utilities to display, sort, and select attributes from a name in the CSL format.

Usage

csl_name(x)

Arguments

x A list of names conforming to the structure outlined in the CSL schema

See Also

<https://citeproc-js.readthedocs.io/en/latest/csl-json/markup.html#name-fields>

cv_document *Output format for vitae*

Description

This output format provides support for including LaTeX dependencies and bibliography entries in extension of the rmarkdown::pdf_document() format.

Usage

cv_document(..., pandoc_args = NULL, pandoc_vars = NULL)

Arguments

... Arguments passed to rmarkdown::pdf_document().
 pandoc_args Additional command line options to pass to pandoc
 pandoc_vars Pandoc variables to be passed to the template.

Index

[awesomecv](#), [2](#)

[bibliography_entries](#), [3](#)

[brief_entries](#), [4](#)

[csl_date](#), [4](#)

[csl_name](#), [5](#)

[cv_document](#), [2](#), [5](#)

[cv_entries \(brief_entries\)](#), [4](#)

[cv_formats \(awesomecv\)](#), [2](#)

[detailed_entries \(brief_entries\)](#), [4](#)

[hyndman \(awesomecv\)](#), [2](#)

[latexcv \(awesomecv\)](#), [2](#)

[moderncv \(awesomecv\)](#), [2](#)

[rmarkdown::pandoc_citeproc_convert\(\)](#),
[3](#)

[twentyseconds \(awesomecv\)](#), [2](#)