

Package ‘codemetar’

February 12, 2018

Type Package

Title Generate 'CodeMeta' Metadata for R Packages

Version 0.1.4

Description The 'Codemeta' Project defines a 'JSON-LD' format for describing software metadata, as detailed at <<https://codemeta.github.io>>. This package provides utilities to generate, parse, and modify 'codemeta.json' files automatically for R packages, as well as tools and examples for working with 'codemeta.json' 'JSON-LD' more generally.

License MIT + file LICENSE

URL <https://github.com/ropensci/codemetar>

BugReports <https://github.com/ropensci/codemetar/issues>

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Depends R (>= 3.0.0)

Imports jsonlite (>= 1.3), jsonld, git2r, devtools, methods, stats, stringi, readr

Suggests testthat, jsonvalidate, covr, knitr, rmarkdown, httr, magrittr, xml2, dplyr (>= 0.7.0), tibble, purrr, printr

VignetteBuilder knitr

X-schema.org-keywords metadata, codemeta, ropensci, citation, credit, linked-data

X-schema.org-isPartOf <https://ropensci.org>

X-schema.org-relatedLink <https://codemeta.github.io/codemetar>

NeedsCompilation no

Author Carl Boettiger [aut, cre, cph]
(<<https://orcid.org/0000-0002-1642-628X>>)

Maintainer Carl Boettiger <cboettig@gmail.com>

Repository CRAN

Date/Publication 2018-02-12 18:29:20 UTC

R topics documented:

codemetar-package	2
add_context	3
codemeta_validate	3
create_codemeta	4
crosswalk	5
crosswalk_table	6
crosswalk_transform	6
drop_context	7
write_codemeta	7

Index	9
--------------	----------

codemetar-package	<i>codemetar: generate codemeta metadata for R packages</i>
-------------------	---

Description

The 'CodeMeta' Project defines a JSON-LD format for describing software metadata, as detailed at <https://codemeta.github.io>. This package provides utilities to generate, parse, and modify codemeta.jsonld files automatically for R packages, as well as tools and examples for working with codemeta JSON-LD more generally.

Details

It has three main goals:

- Quickly generate a valid codemeta.json file from any valid R package To do so, we automatically extract as much metadata as possible using the DESCRIPTION file, as well as extracting metadata from other common best-practices such as the presence of Travis and other badges in README, etc.
- Facilitate the addition of further metadata fields into a codemeta.json file, as well as general manipulation of codemeta files.
- Support the ability to crosswalk between terms used in other metadata standards, as identified by the CodeMeta Project Community, see <https://codemeta.github.io/crosswalk>

To learn more about codemetar, start with the vignettes: `browseVignettes(package = "codemetar")`

For more general information about the CodeMeta Project for defining software metadata, see <https://codemeta.github.io>. In particular, new users might want to start with the User Guide (<https://codemeta.github.io/user-guide/>), while those looking to learn more about JSON-LD and consuming existing codemeta files should see the Developer Guide (<https://codemeta.github.io/developer-guide/>).

Author(s)

Maintainer: Carl Boettiger <cboettig@gmail.com> (0000-0002-1642-628X) [copyright holder]

See Also

Useful links:

- <https://github.com/ropensci/codemeta>
- Report bugs at <https://github.com/ropensci/codemeta/issues>

 add_context

add_context

Description

add context element to json list or json string

Usage

```
add_context(x, context, json_output = FALSE)
```

Arguments

x	a JSON list (from read_json / fromJSON) or json object (from toJSON)
context	context to be added, in same format as x
json_output	logical, should output be a json object or a list?

Value

a list or json object with "@context" element added

 codemeta_validate

codemeta validate

Description

Checks that a round-trip of expanding and compacting json-ld matches the original document. Incorrect schema terms or types (or spelling errors of properties) will cause this round-trip to fail by not fully compacting.

Usage

```
codemeta_validate(codemeta = "codemeta.json", context = NULL)
```

Arguments

codemeta	path/filename to a codemeta.json file, or json-ld text string
context	URL (or path or json string) for the codemeta context. Leave as default or use appropriate DOI for the version; see details.

Details

by default, validation will use the original context from the import file.

Examples

```
ex <- system.file("examples/codemeta.json", package="codemeta")
codemeta_validate(ex)
```

create_codemeta	<i>create_codemeta</i>
-----------------	------------------------

Description

create a codemeta list object in R for further manipulation. Similar to [write_codemeta](#), but returns an R list object rather than writing directly to a file. See examples.

Usage

```
create_codemeta(pkg = ".", root = ".", id = NULL,
  force_update = getOption("codemeta_force_update", TRUE), ...)
```

Arguments

pkg	package path to package root, or package name, or description file (character), or a codemeta object (list)
root	if pkg is a codemeta object, optionally give the path to package root. Default guess is current dir.
id	identifier for the package, e.g. a DOI (or other resolvable URL)
force_update	Update guessed fields even if they are defined in an existing codemeta.json file
...	additional arguments to write_json

Value

a codemeta list object

Examples

```
cm <- create_codemeta("jsonlite")
cm$keywords <- list("metadata", "ropensci")
```

crosswalk	<i>crosswalk</i>
-----------	------------------

Description

Crosswalk between different metadata fields used by different repositories, registries and archives. For more details see <https://codemeta.github.io/crosswalk> This function requires an internet connection to obtain the latest crosswalk table. This function essentially applies the crosswalk table shown by [crosswalk_table](#) to a given JSON metadata record.

Usage

```
crosswalk(x, from, to = "codemeta",
          codemeta_context = getOption("codemeta_context",
          "http://purl.org/codemeta/2.0"))
```

Arguments

x	a JSON list or file with data fields to be crosswalked
from	the corresponding column name from the crosswalk table.
to	the column to translate into, assumes "codemeta" by default
codemeta_context	the address or contents of codemeta context. Leave at default

Value

a 'json' object containing a valid codemeta.json file created by crosswalking the input JSON

Examples

```
## Not run:
## Crosswalk data returned by the GitHub API into CodeMeta format
r <- gh::gh("/repos/:owner/:repo", owner = "ropensci", repo = "EML")
crosswalk(r, "GitHub")

## End(Not run)
```

crosswalk_table	<i>crosswalk_table</i>
-----------------	------------------------

Description

return a subset of the crosswalk table containing codemeta properties and matching column

Usage

```
crosswalk_table(from, to = NULL,
  full_crosswalk = "https://github.com/codemeta/codemeta/raw/master/crosswalk.csv",
  trim = TRUE)
```

Arguments

from	the name of a column in the crosswalk table to map from.
to	the name of one or more columns in the crosswalk table to map into
full_crosswalk	Path or URL (requires internet!) of the full crosswalk table.
trim	drop rows giving properties not found in the 'from' column?

Value

a tibble containing the trimmed crosswalk table, listing property (in CodeMeta), and the corresponding terms in both from and to columns.

Examples

```
## Not run:
crosswalk_table(from = "GitHub", to = c("Zenodo", "Figshare"))

## End(Not run)
```

crosswalk_transform	<i>Crosswalk transform</i>
---------------------	----------------------------

Description

Perform JSON-LD expansion of input, followed by compaction into the codemeta context

Usage

```
crosswalk_transform(x, crosswalk_context = NULL,
  codemeta_context = getOption("codemeta_context",
  "http://purl.org/codemeta/2.0"))
```

Arguments

x a JSON list or file with data fields to be crosswalked
 crosswalk_context Context to be added to x
 codemeta_context the address or contents of codemeta context. Leave at default

Value

a valid codemeta json description.

drop_context	<i>drop_context</i>
--------------	---------------------

Description

drop context element from json list or json string

Usage

```
drop_context(x, json_output = FALSE)
```

Arguments

x a JSON list (from read_json / fromJSON) or json object (from toJSON)
 json_output logical, should output be a json object or a list?

Value

a list or json object with the "@context" element removed

write_codemeta	<i>write_codemeta</i>
----------------	-----------------------

Description

write out a codemeta.json file for a given package. This function is basically a wrapper around create_codemeta() to both create the codemeta object and write it out to a JSON-LD-formatted file in one command. It can also be used simply to write out to JSON-LD any existing object created with create_codemeta().

Usage

```
write_codemeta(pkg = ".", path = "codemeta.json", root = ".", id = NULL,  

  force_update = getOption("codemeta_force_update", TRUE), ...)
```

Arguments

pkg	package path to package root, or package name, or description file (character), or a codemeta object (list)
path	file name of the output, leave at default "codemeta.json"
root	if pkg is a codemeta object, optionally give the path to package root. Default guess is current dir.
id	identifier for the package, e.g. a DOI (or other resolvable URL)
force_update	Update guessed fields even if they are defined in an existing codemeta.json file
...	additional arguments to write_json

Details

If pkg is a codemeta object, the function will attempt to update any fields it can guess (i.e. from the DESCRIPTION file), overwriting any existing data in that block. In this case, the package root directory should be the current working directory.

Value

writes out the codemeta.json file

Examples

```
write_codemeta("codemeta", tempfile())
```


Index

`add_context`, [3](#)

`codemeta_validate`, [3](#)

`codemeta` (`codemeta`-package), [2](#)

`codemeta`-package, [2](#)

`create_codemeta`, [4](#)

`crosswalk`, [5](#)

`crosswalk_table`, [5](#), [6](#)

`crosswalk_transform`, [6](#)

`drop_context`, [7](#)

`write_codemeta`, [4](#), [7](#)

`write_json`, [4](#), [8](#)