# Package 'codemetar'

March 12, 2019

**Type** Package

**Title** Generate 'CodeMeta' Metadata for R Packages

**Version** 0.1.7

**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** GPL-3

**URL** <https://github.com/ropensci/codemetar>, <https://ropensci.github.io/codemetar>

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

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Depends** R (>= 3.0.0)

**Imports** jsonlite (>= 1.6), git2r, pkgbuild, memoise, methods, stats, stringi, readr, desc, usethis, whisker, tibble, crul, gh, stringr, sessioninfo, purrr, curl, commonmark, xml2, magrittr, glue

**Suggests** testthat, jsonld, jsonvalidate, covr, knitr, rmarkdown, dplyr (>= 0.7.0), printr

**VignetteBuilder** knitr

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

**X-schema.org-isPartOf** https://ropensci.org

**NeedsCompilation** no

**Author** Carl Boettiger [aut, cre, cph] (<https://orcid.org/0000-0002-1642-628X>), Anna Krystalli [rev, ctb] (<https://orcid.org/0000-0002-2378-4915>),

Toph Allen [rev] (<https://orcid.org/0000-0003-4580-091X>),
Maëlle Salmon [ctb, aut] (<https://orcid.org/0000-0002-2815-0399>),
rOpenSci [fnd] (https://ropensci.org/),
Katrin Leinweber [ctb] (<https://orcid.org/0000-0001-5135-5758>),
Noam Ross [ctb] (<https://orcid.org/0000-0002-2136-0000>),
Arfon Smith [ctb],
Jeroen Ooms [ctb] (<https://orcid.org/0000-0002-4035-0289>),
Sebastian Meyer [ctb] (<https://orcid.org/0000-0002-1791-9449>),
Michael Rustler [ctb] (<https://orcid.org/0000-0003-0647-7726>),
Hauke Sonnenberg [ctb] (<https://orcid.org/0000-0001-9134-2871>)

**Maintainer** Carl Boettiger <cboettig@gmail.com>

**Repository** CRAN

**Date/Publication** 2019-03-12 19:16:04 UTC

## R topics documented:

---

codemetar-package        *codemetar: generate codemeta metadata for R packages*

---

### 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]

Authors:

- Maëlle Salmon (0000-0002-2815-0399) [contributor]

Other contributors:

- Anna Krystalli (0000-0002-2378-4915) [reviewer, contributor]
- Toph Allen (0000-0003-4580-091X) [reviewer]
- rOpenSci (https://ropensci.org/) [funder]
- Katrin Leinweber (0000-0001-5135-5758) [contributor]
- Noam Ross (0000-0002-2136-0000) [contributor]
- Arfon Smith [contributor]
- Jeroen Ooms (0000-0002-4035-0289) [contributor]
- Sebastian Meyer (0000-0002-1791-9449) [contributor]
- Michael Rustler (0000-0003-0647-7726) [contributor]
- Hauke Sonnenberg (0000-0001-9134-2871) [contributor]

**See Also**

Useful links:

- https://github.com/ropensci/codemetar
- https://ropensci.github.io/codemetar
- Report bugs at https://github.com/ropensci/codemetar/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="codemetar")
codemeta_validate(ex)
```

---

create_codemeta *create_codemeta*

---

## 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,
  use_filesize = TRUE,
  force_update = getOption("codemeta_force_update", TRUE),
  verbose = 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) |
| use_filesize | whether to try adding a filesize by using pkgbuild::build(). |
| force_update | Update guessed fields even if they are defined in an existing codemeta.json file |
| verbose | Whether to print messages indicating opinions e.g. when DESCRIPTION has no URL. See give_opinions. |
| ... | additional arguments to write_json |

## Value

a codemeta list object

## Examples

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

| crosswalk | *Crosswalk* |
|---|---|

#### 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 = NULL)
```

#### 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. If not specified, the default "https://doi.org/10.5063/schema/c 2.0" is used. The default is taken from the option "codemeta_context" and can thus be overridden by setting this option with options(codemeta_context = <your_context_url>). |

#### 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          *Crosswalk Table*

### Description

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

### Usage

```
crosswalk_table(from, to = NULL, full_crosswalk = NULL, 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

```
crosswalk_table(from = "GitHub", to = c("Zenodo", "Figshare"))
```

drop_context          *Drop Context*

### 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](read_json) / [fromJSON](fromJSON)) or json object (from [toJSON](toJSON)) |
| json_output | logical, should output be a json object or a list? |

### Value

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

---

extract_badges          *Extract all badges from Markdown file*

---

### Description

Extract all badges from Markdown file

### Usage

```
extract_badges(path)
```

### Arguments

path            Path to Markdown file

### Value

A tibble with for each badge its text, link and link to its image.

### Examples

```
## Not run:
extract_badges(system.file("examples/README_fakepackage.md", package="codemetar"))

## End(Not run)
```

---

give_opinions          *Function giving opinions about a package*

---

### Description

Function giving opinions about a package

### Usage

```
give_opinions(pkg_path = getwd())
```

### Arguments

pkg_path       Path to the package root

### Value

A data.frame of opinions

---

write_codemeta *write_codemeta*

---

### 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, use_filesize = TRUE,
  force_update = getOption("codemeta_force_update", TRUE),
  use_git_hook = TRUE, verbose = 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) |
| use_filesize | whether to try adding a filesize by using pkgbuild::build(). |
| force_update | Update guessed fields even if they are defined in an existing codemeta.json file |
| use_git_hook | Whether to create a pre-commit hook requiring codemeta.json to be updated when DESCRIPTION is changed. Defaults to TRUE. |
| verbose | Whether to print messages indicating opinions e.g. when DESCRIPTION has no URL. See give_opinions. |
| ... | 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.

When creating and writing a codemeta.json for the first time, the function adds "codemeta.json" to .Rbuildignore and, if the project uses git, adds a pre-commit hook ensuring that if DESCRIPTION changes, the codemeta.json will be updated as well unless the DESCRIPTION change is committed with 'git commit –no-verify'.

## Value

writes out the codemeta.json file

## Examples

```
write_codemeta("codemetar", path = "example_codemetar_codemeta.json")
```

# Index