

Package: rocrateR (via r-universe)

May 22, 2026

Title Tools for Creating and Manipulating RO-Crates

Version 0.1.0.9000

Description Provides tools for creating, manipulating and reading Research Object Crates (RO-Crates), a lightweight approach to packaging research data with structured metadata. Includes utilities for metadata generation, entity management, validation and reading existing RO-Crates following the specification <<https://w3id.org/ro/crate/1.2/>>.

License MIT + file LICENSE

Suggests fs, knitr, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

URL <https://github.com/ResearchObject/ro-crate-r/>

BugReports <https://github.com/ResearchObject/ro-crate-r/issues/>

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports digest, jsonlite, lifecycle, zip

Depends R (>= 4.1.0)

VignetteBuilder knitr

Repository <https://researchobject.r-universe.dev>

Date/Publication 2026-04-22 11:29:29 UTC

RemoteUrl <https://github.com/researchobject/ro-crate-r>

RemoteRef HEAD

RemoteSha 45ddd64ced1ae2d4e29e459239970f9b5b3cf8c

Contents

add_author	2
add_dataset	3
add_entity	4
add_entity_value	5
add_notebook	6
add_project	7
add_readme	7
add_software	8
add_workflow	8
bag_rocrate	9
crate_project	11
entity	11
extract_content	12
get_entity	13
is_rocrate	14
is_rocrate_bag	15
load_rocrate	16
load_rocrate_bag	17
print.entity	18
print.rocrate	19
read_rocrate	20
remove_entity	20
rocrate	21
rocrate_5s	22
summary.entity	24
summary.rocrate	24
unbag_rocrate	25
validate_rocrate	26
write_rocrate	27
Index	28

add_author	<i>Add an author to an RO-Crate</i>
------------	-------------------------------------

Description

This helper creates an author entity and if affiliation is provided, then creates an organisation entity for the user's affiliation.

Usage

```
add_author(  
  rocrate,  
  name,  
  orcid = NULL,  
  affiliation = NULL,  
  ror = NULL,  
  set_author = TRUE  
)
```

Arguments

rocrate	RO-Crate object, see rocrate .
name	Author's name.
orcid	Optional, ORCID identifier, for details see https://orcid.org .
affiliation	Optional, author's organisation.
ror	Optional, ROR identifier for the affiliation, for details see https://ror.org .
set_author	Logical, used to indicate if the current user should be set as the author of the RO-Crate.

Value

Updated RO-Crate object.

add_dataset	<i>Add a dataset to an RO-Crate</i>
-------------	-------------------------------------

Description

This helper converts an R object (typically a `data.frame`) into a dataset inside the RO-Crate. The object is stored in the content field and written to disk when calling `extract_content()` or `bag_rocrate(write_content = TRUE)`.

Usage

```
add_dataset(  
  rocrate,  
  file_id,  
  data = NULL,  
  name = NULL,  
  description = NULL,  
  encodingFormat = "text/csv"  
)
```

Arguments

rocrate	RO-Crate object, see rocrate .
file_id	Filename for the dataset file.
data	R object to store (typically a <code>data.frame</code>).
name	Dataset name.
description	Optional dataset description.
encodingFormat	MIME type (default "text/csv").

Details

Use this when you want to register a dataset file or directory as a formal Dataset entity inside the RO-Crate metadata.

Value

Updated RO-Crate object.

add_entity	<i>Add entity to RO-Crate</i>
------------	-------------------------------

Description

Add entity to RO-Crate

Usage

```
add_entity(rocrate, entity, overwrite = FALSE, verbose = FALSE)
```

```
add_entities(rocrate, entity, overwrite = FALSE, verbose = FALSE)
```

Arguments

rocrate	RO-Crate object, see rocrate .
entity	Entity object (list) that contains at least the following components: @id and @type. Alternatively, a list of entities.
overwrite	Boolean flag to indicate if the entity (if found in the given RO-Crate) should be overwritten.
verbose	Boolean flag to indicate if status messages should be hidden (default: FALSE).

Value

Updated RO-Crate object.

Examples

```

basic_crate <- rocrateR::rocrate()

# create entity for an organisation
organisation_uol <- rocrateR::entity(
  "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)

basic_crate_v2 <- basic_crate |>
  rocrateR::add_entity(person_rvd) |>
  rocrateR::add_entity_value(
    id = "./",
    key = "author",
    value = list(`@id` = person_rvd$`@id`)
  ) |>
  rocrateR::add_entity(organisation_uol)

```

add_entity_value *Add entity value to RO-Crate*

Description

Add entity value to RO-Crate, under entity with @id = {id}, using the pair {key}-{value} within @graph.

Usage

```
add_entity_value(rocrate, id, key, value, overwrite = FALSE, verbose = FALSE)
```

Arguments

rocrate	RO-Crate object, see rocrate .
id	String with the ID of the RO-Crate entity within @graph.
key	String with the key of the entity with id to be modified.
value	String with the value for key.
overwrite	Boolean flag to indicate if the existing value (if any), should be overwritten (default: FALSE).
verbose	Boolean flag to indicate if status messages should be hidden (default: FALSE).

Value

Updated RO-Crate object.

Examples

```
basic_crate <- rocrate()

# create entity for an organisation
organisation_uol <- rocrateR::entity(
  "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)

basic_crate_v2 <- basic_crate |>
  rocrateR::add_entity_value(
    id = "./",
    key = "author",
    value = list(`@id` = person_rvd$`@id`)
  )
```

add_notebook

Add a notebook to an RO-Crate

Description

Add a notebook to an RO-Crate

Usage

```
add_notebook(rocrate, file_id, name = NULL, content = NULL)
```

Arguments

rocrate	RO-Crate object, see rocrate .
file_id	Optional, notebook's filename.
name	Optional, notebook's name.
content	Optional, notebook's content.

Value

Updated RO-Crate object.

add_project	<i>Add project metadata</i>
-------------	-----------------------------

Description

Add project metadata

Usage

```
add_project(rocrate, name, description = NULL)
```

Arguments

rocrate	RO-Crate object, see rocrate .
name	Project's name.
description	Optional, project's description.

Value

Updated RO-Crate object.

add_readme	<i>Add README file to an RO-Crate</i>
------------	---------------------------------------

Description

Add README file to an RO-Crate

Usage

```
add_readme(rocrate, text, filename = "README.md")
```

Arguments

rocrate	RO-Crate object, see rocrate .
text	Character vector with README content.
filename	README filename.

Value

Updated RO-Crate object.

add_software	<i>Add software application entity</i>
--------------	--

Description

Add software application entity

Usage

```
add_software(rocrate, name, version = NULL)
```

Arguments

rocrate	RO-Crate object, see rocrate .
name	Software name.
version	Version string.

Value

Updated RO-Crate object.

add_workflow	<i>Add a workflow to an RO-Crate</i>
--------------	--------------------------------------

Description

Register a workflow script (e.g. R, Python, Nextflow) as a ComputationalWorkflow entity inside the RO-Crate.

Usage

```
add_workflow(  
  rocrate,  
  file_id,  
  name = NULL,  
  description = NULL,  
  language = "R",  
  content = NULL  
)
```

Arguments

rocrate	RO-Crate object, see rocrate .
file_id	Filename of the workflow script.
name	Workflow name.
description	Optional description.
language	Programming language (default "R").
content	Optional script contents.

Value

Updated RO-Crate object.

bag_rocrate	<i>Bag the contents of an RO-Crate</i>
-------------	--

Description

Bag the contents of an RO-Crate using the BagIt file packaging format v1.0. For more details see the definition: [doi:10.17487/RFC8493](https://doi.org/10.17487/RFC8493)

Usage

```
bag_rocrate(x, ...)  
  
## S3 method for class 'character'  
bag_rocrate(x, ..., output = x, force_bag = FALSE, extra_bag_info = NULL)  
  
## S3 method for class 'rocrate'  
bag_rocrate(  
  x,  
  ...,  
  path,  
  output = path,  
  overwrite = FALSE,  
  force_bag = FALSE,  
  extra_bag_info = NULL,  
  write_content = TRUE,  
  create_dir = TRUE  
)
```

Arguments

x	A string to a path containing at the very minimum an RO-Crate metadata descriptor file, <code>ro-crate-metadata.json</code> . Alternatively, an object with the <code>rocrate</code> class.
...	Additional parameters, see below.
output	String with path where the RO-Crate bag will be stored (default: x - same path as the input value).
force_bag	Boolean flag to indicate whether to force the creation of a 'bag' even if not all the files were successfully bagged (default: FALSE ~ check if all the files were copied successfully).
extra_bag_info	Vector of strings to include in the <code>bag-info.txt</code> file (e.g., <code>Contact-Email: first.last@rocrate.org</code>).
path	String with path to the root of the RO-Crate.
overwrite	Boolean flag to indicate if the RO-Crate metadata descriptor file should be overwritten if already inside path (default: FALSE).
write_content	Logical. If TRUE, write content fields of File entities to disk before bagging.
create_dir	Boolean flag to indicate if the path should be created, if it doesn't exist.

Value

String with full path to the RO-Crate bag.

See Also

Other RO-Crate BagIt archive functions: `is_rocrate_bag()`, `load_rocrate_bag()`, `unbag_rocrate()`

Examples

```
# ----- SETUP -----
basic_crate <- rocrateR::rocrate()
# temp file
tmp_dir <- file.path(tempdir(), digest::digest(basename(tempfile())))
tmp <- file.path(tmp_dir, "ro-crate-metadata.json")
dir.create(tmp_dir)

# ----- INPUT: RO-Crate -----
rocrateR::bag_rocrate(basic_crate, path = tmp_dir)

# ----- INPUT: Path -----
rocrateR::bag_rocrate(tmp_dir, output = tmp_dir)

# delete temp directory
unlink(tmp_dir, recursive = TRUE)
```

crate_project	<i>Automatically create an RO-Crate from a project directory</i>
---------------	--

Description

Automatically create an RO-Crate from a project directory

Usage

```
crate_project(path = NULL)
```

Arguments

path	String with project directory.
------	--------------------------------

Value

RO-Crate object for the project given by path.

entity	<i>Create a data entity</i>
--------	-----------------------------

Description

Create a data entity

Usage

```
entity(id, type, ...)
```

Arguments

id	Scalar value with @id for the entity (e.g., character, numeric).
type	String with @type for the entity (e.g., Dataset, File).
...	Optional additional entity values/properties.

Value

List with an entity object.

Examples

```
# create entity for an organisation
organisation_uol <- rocrateR::entity(
  "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)
```

extract_content

Extract File entities content to files

Description

Write the content field of File entities to disk using their @id as the filename.

Usage

```
extract_content(rocrate, path, overwrite = FALSE)
```

Arguments

rocrate	RO-Crate object, see rocrate .
path	Directory where files will be written. RO-Crate root.
overwrite	Logical. Overwrite existing files.

Value

Invisibly returns updated rocrate without contents.

get_entity	<i>Get entity(ies)</i>
------------	------------------------

Description

Get entity(ies)

Usage

```
get_entity(rocrate, id = NULL, type = NULL)
```

Arguments

rocrate	RO-Crate object, see rocrate .
id	String with the ID of the RO-Crate entity within @graph (optional if type is provided). Alternatively, an entity object / list with @id and @type.
type	String with the type of the RO-Crate entity(ies) within @graph to retrieve (optional if id is provided).

Value

List with found entity object(s), if any, NULL otherwise.

Examples

```
basic_crate <- rocrateR::rocrate()

# create entity for an organisation
organisation_uol <- rocrateR::entity(
  "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)

basic_crate_person <- basic_crate |>
  rocrateR::add_entity(person_rvd) |>
  rocrateR::add_entity_value(
    id = "./",
    key = "author",
    value = list(`@id` = person_rvd$`@id`)
```

```

) |>
rocrateR::add_entity(organisation_uol) |>
rocrateR::get_entity(person_rvd)

basic_crate_person[[1]]$name == person_rvd$name
basic_crate_person[[1]]$`@id` == person_rvd$`@id`

```

is_rocrate	<i>Check if object is an RO-Crate</i>
------------	---------------------------------------

Description

Check if object is an RO-Crate

Usage

```
is_rocrate(rocrate, strict = FALSE, error = TRUE)
```

Arguments

rocrate	RO-Crate object, see rocrate .
strict	Boolean to indicate if stricter checks should be done (e.g., check profile specification).
error	Boolean to indicate if the function should throw an error, if any errors are found (default: TRUE).

Value

Boolean flag with RO-Crate validity.

Examples

```

basic_crate <- rocrateR::rocrate()

# check if the new crate is valid
basic_crate |>
  rocrateR::is_rocrate()

```

is_rocrate_bag	<i>Check if path points to a valid RO-Crate bag</i>
----------------	---

Description

Check if path points to a valid RO-Crate bag

Usage

```
is_rocrate_bag(path, algo = NULL, bagit_version = "1.0")
```

Arguments

path	String with full path to a compressed file contain an RO-Crate bag, see bag_rocrate for details. Alternatively, a path to a directory containing an RO-Crate bag.
algo	String with algorithm used to generate the RO-Crate bag (default: NULL, which auto detects the algorithm from the manifest-<algo>.txt file, inside the bag). See digest for more details on valid algorithms.
bagit_version	String with version of BagIt used to generate the RO-Crate bag (default: "1.0"). See doi:10.17487/RFC8493 for more details.

Value

Returns a boolean flag to indicate if the given RO-Crate bag is valid.

See Also

Other RO-Crate BagIt archive functions: [bag_rocrate\(\)](#), [load_rocrate_bag\(\)](#), [unbag_rocrate\(\)](#)

Examples

```
# ----- SETUP -----
basic_crate <- rocrateR::rocrate()
# temp file
tmp_dir <- file.path(tempdir(), digest::digest(basename(tempfile()))))
tmp <- file.path(tmp_dir, "ro-crate-metadata.json")
dir.create(tmp_dir)

# bag RO-Crate
path_to_roc_bag <- rocrateR::bag_rocrate(basic_crate, path = tmp_dir)

# ----- INPUT: RO-Crate BagIt archive -----
rocrateR::is_rocrate_bag(path_to_roc_bag)

# ----- INPUT: Path -----
rocrateR::unbag_rocrate(path_to_roc_bag) |>
  rocrateR::is_rocrate_bag()
```

```
# delete temp directory
unlink(tmp_dir, recursive = TRUE)
```

load_rocrate	<i>Load an RO-Crate from various input types</i>
--------------	--

Description

High-level loader that can read:

- A ro-crate-metadata.json file
- A directory containing an RO-Crate
- A BagIt-wrapped RO-Crate directory
- A zipped BagIt RO-Crate archive

Usage

```
load_rocrate(x, ...)

## S3 method for class 'rocrate'
load_rocrate(x, ..., verbose = FALSE)

## S3 method for class 'character'
load_rocrate(
  x,
  ...,
  verbose = FALSE,
  bagit_version = "1.0",
  load_content = FALSE,
  max_file_size = 10 * 1024^2
)
```

Arguments

x	A path (character) or an existing rocrate object.
...	Reserved for future extensions.
verbose	Logical. If TRUE, emit diagnostic messages.
bagit_version	String with version of BagIt used to generate the RO-Crate bag (default: "1.0"). See doi:10.17487/RFC8493 for more details.
load_content	Logical. If TRUE, attempt to load external file contents into the content field for entities of type File.
max_file_size	Maximum file size (bytes) allowed when loading content. Default 10MB.

Value

An RO-Crate object.

Examples

```

# ----- SETUP -----
basic_crate <- rocrateR::rocrate()
# temp file
tmp_dir <- file.path(tempdir(), digest::digest(basename(tempfile()))))
tmp <- file.path(tmp_dir, "ro-crate-metadata.json")
dir.create(tmp_dir)

# ----- INPUT: RO-Crate -----
rocrateR::load_rocrate(basic_crate, verbose = TRUE)

# ----- INPUT: Path -----
# save RO-Crate
rocrateR::write_rocrate(basic_crate, path = tmp)

# load RO-Crate
## with file name
rocrateR::load_rocrate(tmp, verbose = TRUE)

## with directory
rocrateR::load_rocrate(tmp_dir, verbose = TRUE)

# delete temp directory
unlink(tmp_dir, recursive = TRUE)

```

load_rocrate_bag	<i>Load an RO-Crate BagIt archive</i>
------------------	---------------------------------------

Description

Load an RO-Crate BagIt archive

Usage

```

load_rocrate_bag(
  path,
  algo = NULL,
  bagit_version = "1.0",
  load_content = FALSE,
  max_file_size = 10 * 1024^2
)

```

Arguments

path	String with full path to a compressed file contain an RO-Crate bag, see bag_rocrate for details. Alternatively, a path to a directory containing an RO-Crate bag.
algo	String with algorithm used to generate the RO-Crate bag (default: NULL, which auto detects the algorithm from the manifest-<algo>.txt file, inside the bag). See digest for more details on valid algorithms.

<code>bagit_version</code>	String with version of BagIt used to generate the RO-Crate bag (default: "1.0"). See doi:10.17487/RFC8493 for more details.
<code>load_content</code>	Logical. If TRUE, attempt to load external file contents into the content field for entities of type File.
<code>max_file_size</code>	Maximum file size (bytes) allowed when loading content. Default 10MB.

Value

An object with the `rocrate` class.

See Also

Other RO-Crate BagIt archive functions: `bag_rocrate()`, `is_rocrate_bag()`, `unbag_rocrate()`

Examples

```
# ----- SETUP -----
basic_crate <- rocrateR::rocrate()
# temp file
tmp_dir <- file.path(tempdir(), digest::digest(basename(tempfile()))))
tmp <- file.path(tmp_dir, "ro-crate-metadata.json")
dir.create(tmp_dir)

# bag RO-Crate
path_to_roc_bag <- rocrateR::bag_rocrate(basic_crate, path = tmp_dir)

# ----- INPUT: RO-Crate BagIt archive -----
rocrateR::load_rocrate_bag(path_to_roc_bag)

# ----- INPUT: Path -----
rocrateR::unbag_rocrate(path_to_roc_bag) |>
  rocrateR::load_rocrate_bag()

# delete temp directory
unlink(tmp_dir, recursive = TRUE)
```

`print.entity`

Print RO-Crate entity

Description

Print RO-Crate entity, S3 method for class 'entity'.

Usage

```
## S3 method for class 'entity'
print(x, ...)
```

Arguments

x RO-Crate entity object, see [entity](#).
 ... Optional arguments, not used.

Value

Invisibly the input RO-Crate entity, x.

Examples

```
rocrateR::rocrate() |>
  rocrateR::get_entity("./")
```

print.rocrate	<i>Print RO-Crate</i>
---------------	-----------------------

Description

Print RO-Crate, S3 method for class 'rocrate'. Creates a temporal JSON file, which then is displayed with the [message](#) function.

Usage

```
## S3 method for class 'rocrate'
print(x, ..., max_lines = getOption("max_lines", 100))
```

Arguments

x RO-Crate object, see [rocrate](#).
 ... Optional arguments, not used.
 max_lines Max number of lines to display.

Value

Invisibly the input RO-Crate, x.

Examples

```
rocrateR::rocrate()
```

read_rocrate	<i>Wrapper for <code>jsonlite::read_json</code></i>
--------------	---

Description

Wrapper for `jsonlite::read_json`. Enforces that the object read is an RO-Crate.

Usage

```
read_rocrate(path, simplifyVector = FALSE, ...)
```

Arguments

<code>path</code>	file on disk
<code>simplifyVector</code>	simplifies nested lists into vectors and data frames. See <code>fromJSON()</code> .
<code>...</code>	Arguments passed on to <code>jsonlite::fromJSON</code>
<code>txt</code>	a JSON string, URL or file
<code>simplifyDataFrame</code>	coerce JSON arrays containing only records (JSON objects) into a data frame
<code>simplifyMatrix</code>	coerce JSON arrays containing vectors of equal mode and dimension into matrix or array
<code>flatten</code>	automatically <code>flatten()</code> nested data frames into a single non-nested data frame

Value

Invisibly the RO-Crate stored in `path`.

remove_entity	<i>Remove entity</i>
---------------	----------------------

Description

Remove entity

Usage

```
remove_entity(rocrate, entity, verbose = FALSE)
```

```
remove_entities(rocrate, entity, verbose = TRUE)
```

Arguments

rocrate	RO-Crate object, see rocrate .
entity	Entity object (list) that contains at least the following components: @id and @type. Alternatively, a list of entities.
verbose	Boolean flag to indicate if status messages should be hidden (default: FALSE).

Value

Updated RO-Crate object.

Examples

```
basic_crate <- rocrateR::rocrate()

# create entity for an organisation
organisation_uol <- rocrateR::entity(
  "https://ror.org/04xs57h96",
  type = "Organization",
  name = "University of Liverpool",
  url = "http://www.liv.ac.uk"
)

# create an entity for a person
person_rvd <- rocrateR::entity(
  "https://orcid.org/0000-0001-5036-8661",
  type = "Person",
  name = "Roberto Villegas-Diaz",
  affiliation = list(`@id` = organisation_uol$`@id`)
)

basic_crate_v2 <- basic_crate |>
  rocrateR::add_entity(person_rvd) |>
  rocrateR::add_entity_value(
    id = "./",
    key = "author",
    value = list(`@id` = person_rvd$`@id`)
  ) |>
  rocrateR::add_entity(organisation_uol) |>
  rocrateR::remove_entity(person_rvd)
```

rocrate

Create a new RO-Crate object

Description

Create a new RO-Crate object. This object includes basic skeleton for the RO-Crate metadata descriptor (ro-crate-metadata.json) file, as described in the official documentation: <https://w3id.org/ro/crate/1.2/>
> **Root Data Entity**.

Usage

```
rocrate(
  ...,
  context = "https://w3id.org/ro/crate/1.2/context",
  conformsTo = gsub("\\/context$", "\\1", context),
  datePublished = Sys.Date(),
  description = "",
  license = "http://spdx.org/licenses/CC-BY-4.0",
  name = ""
)
```

Arguments

...	Optional entities to include in the RO-Crate (e.g., author).
context	String with URL to the version of the RO-Crate specification to use. The context brings the defined terms into the metadata document (default: https://w3id.org/ro/crate/1.2/context).
conformsTo	String with URL to the version of the RO-Crate specification which this object conforms to. Conformance declares which RO-Crate conventions of using those terms are being followed (default: URL formed by context/context)
datePublished	String (or Date object) with the date in which the RO-Crate was published (default: current date).
description	String with description for the root entity (default: empty string).
license	String with URL (permalinks are preferred, but not required) to license to be used for the overall RO-Crate. See the following resources for license choices: https://spdx.org/licenses and/or https://github.com/spdx/license-list-data/tree/main/jsonld (default: CC-BY-4.0: Creative Commons Attribution 4.0 International).
name	String with a name/title for the root entity (default: empty string).

Value

RO-Crate object, list with an additional class, rocrate.

Examples

```
rocrateR::rocrate()
```

rocrate_5s

Create a new 5 Safes RO-Crate object

Description

Create a new 5 Safes RO-Crate object. This object includes basic skeleton for the RO-Crate metadata descriptor (ro-crate-metadata.json) file, as described in the official documentation: <https://w3id.org/ro/crate/1.2> > **Root Data Entity**. Additionally, it includes a profile for the 5 Safes RO-Crate: <https://w3id.org/5s-crate/0.4>

Usage

```

rocrate_5s(
  ...,
  context = "https://w3id.org/ro/crate/1.2/context",
  conformsTo = gsub("\\context$", "\\1", context),
  datePublished = Sys.Date(),
  description = "",
  license = "http://spdx.org/licenses/CC-BY-4.0",
  name = "",
  v5scrate = 0.4
)

```

Arguments

...	Optional entities to include in the RO-Crate (e.g., author).
context	String with URL to the version of the RO-Crate specification to use. The context brings the defined terms into the metadata document (default: https://w3id.org/ro/crate/1.2/context).
conformsTo	String with URL to the version of the RO-Crate specification which this object conforms to. Conformance declares which RO-Crate conventions of using those terms are being followed (default: URL formed by context/context)
datePublished	String (or Date object) with the date in which the RO-Crate was published (default: current date).
description	String with description for the root entity (default: empty string).
license	String with URL (permalinks are preferred, but not required) to license to be used for the overall RO-Crate. See the following resources for license choices: https://spdx.org/licenses and/or https://github.com/spdx/license-list-data/tree/main/jsonld (default: CC-BY-4.0: Creative Commons Attribution 4.0 International).
name	String with a name/title for the root entity (default: empty string).
v5scrate	Numeric value with the version of the 5 Safes RO-Crate profile to use.

Value

5 Safes RO-Crate object, list with an additional class, rocrate.

Examples

```
rocrateR::rocrate_5s()
```

summary.entity	<i>Summary of an RO-Crate entity</i>
----------------	--------------------------------------

Description

Provides a summary of a single RO-Crate entity.

Usage

```
## S3 method for class 'entity'
summary(object, ...)
```

Arguments

object	An entity object (list).
...	Additional arguments (unused).

Value

An object of class "summary.entity".

summary.rocrate	<i>Summary of an RO-Crate</i>
-----------------	-------------------------------

Description

Provides a summary of the contents of an RO-Crate, including the number of entities, files and basic metadata information.

Usage

```
## S3 method for class 'rocrate'
summary(object, ...)
```

Arguments

object	An object of class "rocrate".
...	Additional arguments (unused).

Value

An object of class "summary.rocrate".

unbag_rocrate	<i>'Unbag' (extract) RO-Crate packed with BagIt</i>
---------------	---

Description

'Unbag' (extract) RO-Crate packed with BagIt

Usage

```
unbag_rocrate(path, output = dirname(path), quiet = FALSE)
```

Arguments

path	String with path to compressed file containing an RO-Crate bag.
output	String with target path where the contents will be extracted (default: <code>dirname(path)</code> - same directory as input path).
quiet	Boolean flag to indicate if messages should be suppressed (default: <code>FALSE</code> - display messages).

Value

String with path to root of the RO-Crate.

See Also

Other RO-Crate BagIt archive functions: [bag_rocrate\(\)](#), [is_rocrate_bag\(\)](#), [load_rocrate_bag\(\)](#)

Examples

```
# ----- SETUP -----
basic_crate <- rocrateR::rocrate()
# temp file
tmp_dir <- file.path(tempdir(), digest::digest(basename(tempfile()))))
tmp <- file.path(tmp_dir, "ro-crate-metadata.json")
dir.create(tmp_dir)

# bag RO-Crate
path_to_roc_bag <- rocrateR::bag_rocrate(basic_crate, path = tmp_dir)

# ----- INPUT: Path -----
rocrateR::unbag_rocrate(path_to_roc_bag)

# delete temp directory
unlink(tmp_dir, recursive = TRUE)
```

validate_rocrate	<i>Validate an RO-Crate</i>
------------------	-----------------------------

Description

Performs structural, semantic and profile validation.

Usage

```
validate_rocrate(x, mode = c("stop", "report"), strict = FALSE)
```

Arguments

x	A path (character) or an existing rocrate object.
mode	Either "stop" or "report".
strict	Logical. Enable profile validation.

Value

A `rocrate_validation` object (in report mode).

Examples

```
# ----- SETUP -----
basic_crate <- rocrateR::rocrate()
# temp file
tmp_dir <- file.path(tempdir(), digest::digest(basename(tempfile()))))
tmp <- file.path(tmp_dir, "ro-crate-metadata.json")
dir.create(tmp_dir)

# ----- INPUT: RO-Crate -----
rocrateR::validate_rocrate(basic_crate)

# ----- INPUT: Path -----
# save RO-Crate
rocrateR::write_rocrate(basic_crate, path = tmp)

## with file name
rocrateR::validate_rocrate(tmp)

## with directory
rocrateR::validate_rocrate(tmp_dir)

# ----- INPUT: Invalid RO-Crate -----
structure(list(), class = "rocrate") |>
  rocrateR::validate_rocrate(mode = "report")

# delete temp directory
unlink(tmp_dir, recursive = TRUE)
```

write_rocrate	<i>Wrapper for <code>jsonlite::write_json</code></i>
---------------	--

Description

Wrapper for `jsonlite::write_json`. Enforces that the input object is an RO-Crate.

Usage

```
write_rocrate(x, path, ...)
```

Arguments

x	RO-Crate object, see rocrate .
path	file on disk
...	Arguments passed on to <code>jsonlite::toJSON</code>
dataframe	how to encode data.frame objects: must be one of 'rows', 'columns' or 'values'
matrix	how to encode matrices and higher dimensional arrays: must be one of 'rowmajor' or 'columnmajor'.
Date	how to encode Date objects: must be one of 'ISO8601' or 'epoch'
POSIXt	how to encode POSIXt (datetime) objects: must be one of 'string', 'ISO8601', 'epoch' or 'mongo'
factor	how to encode factor objects: must be one of 'string' or 'integer'
complex	how to encode complex numbers: must be one of 'string' or 'list'
raw	how to encode raw objects: must be one of 'base64', 'hex' or 'mongo'
null	how to encode NULL values within a list: must be one of 'null' or 'list'
na	how to print NA values: must be one of 'null' or 'string'. Defaults are class specific
digits	max number of decimal digits to print for numeric values. Use <code>I()</code> to specify significant digits. Use <code>NA</code> for max precision.
force	unclass/skip objects of classes with no defined JSON mapping

Value

Invisibly the input RO-Crate, x.

Index

* RO-Crate BagIt archive functions

- bag_rocrate, 9
 - is_rocrate_bag, 15
 - load_rocrate_bag, 17
 - unbag_rocrate, 25
- add_author, 2
- add_dataset, 3
- add_entities (add_entity), 4
- add_entity, 4
- add_entity_value, 5
- add_notebook, 6
- add_project, 7
- add_readme, 7
- add_software, 8
- add_workflow, 8
- bag_rocrate, 9, 15, 17, 18, 25
- crate_project, 11
- digest, 15, 17
- entity, 11, 19
- extract_content, 12
- flatten(), 20
- fromJSON(), 20
- get_entity, 13
- I(), 27
- is_rocrate, 14
- is_rocrate_bag, 10, 15, 18, 25
- jsonlite::fromJSON, 20
- jsonlite::read_json, 20
- jsonlite::toJSON, 27
- jsonlite::write_json, 27
- load_rocrate, 16
- load_rocrate_bag, 10, 15, 17, 25
- message, 19
- print.entity, 18
- print.rocrate, 19
- read_rocrate, 20
- remove_entities (remove_entity), 20
- remove_entity, 20
- rocrate, 3–10, 12–14, 16, 18, 19, 21, 21, 26, 27
- rocrate_5s, 22
- summary.entity, 24
- summary.rocrate, 24
- unbag_rocrate, 10, 15, 18, 25
- validate_rocrate, 26
- write_rocrate, 27